UNIVERSIDAD DE GUAYAQUIL

Facultad de Ciencias Matemáticas y Físicas

Carrera de Ingeniería en Sistemas Computacionales

“Sistema Académico Integrado Módulo: Apariencia Dinámica Web“

TESIS DE GRADO

Previo a la Obtención del Título de :

INGENIERO EN SISTEMAS COMPUTACIONALES

Autores:

Marjorie Tatiana Arias Domínguez
Julia Elizabeth Macías Tulcán
Marcos Alfredo Maldonado Cabrera

GUAYAQUIL – ECUADOR

AÑO: 2006

AGRADECIMIENTO

Nuestros más sinceros agradecimientos a Dios por todas las gracias recibidas, a la Universidad de Guayaquil Carrera de Ingeniería en sistemas Computacionales y a todos sus directivos y profesores por los conocimientos adquiridos dentro de esta noble institución, a nuestra familia por ser quienes durante estos años nos han apoyado para seguir adelante en el logro de este sueño.

DEDICATORIA

Dedicamos este trabajo fruto de la dedicación y esfuerzo de todos estos años a nuestras familias que siempre confiaron en nosotros y acompañaron con amor todas las etapas que vivimos para alcanzar un presente lleno de éxitos y anhelos de triunfos y superación

TRIBUNAL DE GRADUACION

Ing. Fernando Abad M. **Decano**	**Ing.Juan Sánchez H.** **Director de Tesis**
Ing. Víctor Pazmiño M. **Vocal**	**Ing. Moisés Vera** **Vocal**

Ab. Juan Chávez A.
Secretario

DECLARACION EXPRESA

“La autoría de la tesis de grado corresponde exclusivamente a los suscritos, perteneciendo a la Universidad de Guayaquil los derechos que generen la aplicación de la misma”

(Reglamento de Graduación de la Carrera de Ingeniería en Sistemas Computacionales, Art.26).

Tatiana Arias	Julia Macías	Marcos Maldonado

INDICE GENERAL

CAPÍTULO 1

1. MANUAL TECNICO

1.1. FLUJO DE PROCESO DE VENTANAS

1.1.1. Control del Usuario

Nombre: Usuario

Actores: Estudiantes – Profesores - Administradores

Función: Dar acceso al sistema si el usuario lo tiene permitido.

Descripción: El usuario al digitar su login y password puede ingresar a las opciones del Sistema Académico, en nuestro caso, permite darle la opción de cambio de apariencia. La validación del usuario y password realiza una búsqueda en la tabla usuarios de la B.D. “académico”, únicamente

considerando que sea un usuario registrado, no se consideró sus roles o perfiles, ya que el cambio de apariencia es una opción abierta para todo tipo de usuario. El código debe estar registrado en la base de datos para su respectiva verificación.

FORMATOS DE EVENTOS

Evento: Respuesta del Sistema

1. Ingreso de login y password
2. Respuesta Failed…línea 1
3. Verificación en la Base de Datos: tabla usuarios
4. Ejecución de los Bean de Datos: Valida Usuario – Persona – PersonaBO – PersonaForm.
5. Ingreso a la pantalla de Bienvenida: con la plantilla que le da el sistema a todos los usuarios que es la Estándar, caso contrario mostrará la apariencia que haya guardado como una plantilla de usuario.

1.1.2. Control de Bienvenida

Nombre: Bienvenida

Actores: Estudiantes – Profesores - Administradores

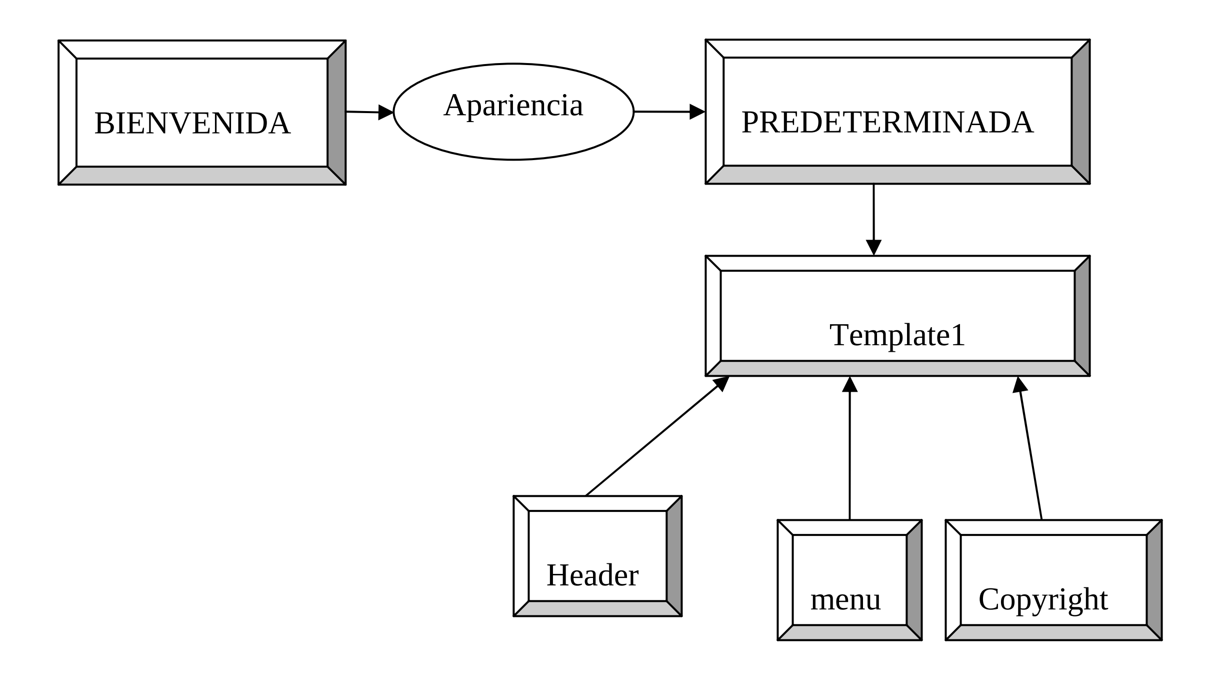
Función: Dar acceso las opciones del sistema incluida el de la Apariencia: (mostrar el menú de opciones)

Descripción: El usuario tendrá la opción de cambiar la apariencia mediante 2 opciones como son la plantilla y el personalizado.

FORMATOS DE EVENTOS

Evento: Respuesta de Personalizar

1. Ingreso a la opción de personalizar plantilla.
2. Respuesta Failed…línea 1
3. Pantalla de EstiloPropio
4. Ejecución de los Bean de Datos: algunos.
5. Cambio de Apariencia.



1.2. DESCRIPCION DE LAS PANTALLAS.

USUARIO.JSP

Descripción: Permitir al usuario el ingreso al sistema, mediante un user y password ya registrado.

Enlaces: Template1, ValidaUsuario, struts-config.xml

Librerias: Taglib de Tags – Bean – Html

Invocación de librerías de Struts.

```
<%@ page contentType="text/html;charset=windows-1252"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-tiles" prefix="tiles"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-bean" prefix="bean"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-html" prefix="html"%>
```

Código que se reutiliza en todas las páginas JSP del Proyecto, para que conserven el mismo estilo de la Plantilla

```
<%@include file="/template.jsp"%>
<tiles:insert beanName="template.default" beanScope="request">
<tiles:put name="body-content" type="string" >
<div id="formulario">
```

Invocar al action para validar al Usuario que entra a la Sesión

```
<html:form action="validaUsuario.do" method="post">
 <table class="apariencia" cellspacing="2" cellpadding="3" border="0" height="150">
      <tr>
        <td width="40%">
            …
          <tr>  <th align="right" bgcolor="Navy">
          <font color="White"><bean:message key="prompt.persona.usuario"/></font>
          </th> <td> <div align="left"> <html:text property="usuario" size="15"/>
            …
           <font color="White"><bean:message key="prompt.persona.clave"/></font>
           </th>  <td>  <html:password property="clave" size="15"/>  </td>
           </tr>  <tr>   <td colspan="2" align="center">
           <html:submit value="Ingresar"/>
           </td>  </tr>  </table></div>   </td>  </tr> </table>  </html:form></div>
</tiles:put></tiles:insert>
```

TEMPLATE.JSP

Descripción: Permitir crear un contenedor donde va ir los componentes visuales, como son el header, menú, copyright. Este template se utiliza en las páginas que no han iniciado una sesión

Enlaces: Header, Menu, Copyright

Librerias: Taglib tiles

```
<%@ page contentType="text/html;charset=windows-1252"%>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-tiles" prefix="tiles"%>

Definir el identifcador del Template y que el alcance será en el  request

<tiles:definition id="template.default" page="layoutPrincipal.jsp" scope="request">
        <tiles:put name="header" value="header.jsp"/>
        <tiles:put name="menubar" value="menu1A.jsp"/>
        <tiles:put name="copyright" value="copyright.jsp"/>
</tiles:definition>
```

HEADER.JSP

Descripción: Cabecera de presentación principal. Contiene un script que muestra la fecha.

Enlaces: insertado en defaultLayout.jsp

Librerias: Ninguna

```
.....
Bean de persona (usuario) que ingreso a la sesión
<jsp:useBean id="unaPersona" class="com.bvg.bean.Persona" scope ="session"/>

...
JavaScript para mostrar el día y fecha actual
<script language=JavaScript>
<!--mydate = new Date();  myday = mydate.getDay();
mymonth = mydate.getMonth(); myear = mydate.getYear();
myweekday= mydate.getDate(); weekday= myweekday;
if(myday == 0) day = " DOMINGO "  else if(myday == 1) day = " LUNES "
else if(myday == 2) day = " MARTES "  else if(myday == 3) day = " MIERCOLES "
else if(myday == 4) day = " JUEVES "  else if(myday == 5) day = " VIERNES "
else if(myday == 6) day = " SABADO "  if(mymonth == 0) month = "ENERO "
else if(mymonth ==1) month = "FEBRERO "  else if(mymonth ==2)
month = "MARZO "  else if(mymonth ==3) month = "ABRIL "  else if(mymonth ==4)
month = "MAYO "  else if(mymonth ==5) month = "JUNIO "  else if(mymonth ==6)
month = "JULIO "  else if(mymonth ==7) month = "AGOSTO "  else if(mymonth ==8)
month = "SEPTIEMBRE " else if(mymonth ==9) month = "OCTUBRE "
else if(mymonth ==10) month = "NOVIEMBRE "  else if(mymonth ==11)
month = "DICIEMBRE "  document.write("<font face=Verdana, Helvetica, sans-serif size=2,
color=#FFFFFF,style=Bold>"+ day); document.write(myweekday+" DE "+month+ " "+myear); // -->
</script>
....
</td>      </tr></table>
```



MENU.JSP

Descripción: Menú Vertical de opciones. Contiene un script que muestra las opciones con que puedes acceder al sistema.

Enlaces: a los modulos BVirtual, Forum, Alumnos, etc.

Librerias: Ninguna

Script que permite el dinamismo del menú Vertical

```
...
<%@ page contentType="text/html;charset=windows-1252"%>
<script language="JavaScript">
<!-- function MM_findObj(n, d) { //v4.01
...
}
function MM_nbGroup(event, grpName) { //v6.0
...
}
//--> </script>
```

Script de la Marquesina deslizante

```
function intializemarquee(){
                    document.cmarquee01.document.cmarquee02.document.write(marqueecontents)
document.cmarquee01.document.cmarquee02.document.close()
                  thelength=document.cmarquee01.document.cmarquee02.document.height
scrollit()                    }         function     scrollit(){                                     if
(document.cmarquee01.document.cmarquee02.top>=thelength*(-1)){
                    document.cmarquee01.document.cmarquee02.top-=speed
                    setTimeout("scrollit()",100)}else{
document.cmarquee01.document.cmarquee02.top=marqueeheight
                    scrollit()       }     }
                    window.onload=regenerate2  </script>     </td>
                          ...
```

COPYRIGHT.JSP

Descripción: Pie de presentación. Contiene un mensaje de pie de página.

Enlaces: Ninguno

Librerias: Ninguna

Vinculos a páginas de información del sistema que están en el pie de página

```
<table id="footer">
<tr>
<td>
<img src="Imagenes/copyright.GIF" width="18" height="21" alt="" />  <a href="#">Acerca de </a> | <a
href="politica.jsp">    Politica de Privacidad</a> | <a href="contacto.jsp">contactenos</a>&copy;2006
Universidad de Guayaquil</td></tr></table>
```

BARRA.JSP

Descripción: Frame que contiene el menú de las Opciones de Plantilla y Personalizar.

Enlaces: insertado en todas las páginas JSP de módulo apariencia

Librerías: Ninguna

```
…

...

var browser = new Browser();

var activeButton = null;
```

Captura y activa o desactiva el botón.

```
if (browser.isIE)
  document.onmousedown = pageMousedown;
else    document.addEventListener("mousedown", pageMousedown, true);
function pageMousedown(event) {
  var el;   if (activeButton == null)
    return;   if (browser.isIE)   el = window.event.srcElement;
  else    el = (event.target.tagName ? event.target : event.target.parentNode);
  if (el == activeButton)   return;  if (getContainerWith(el, "DIV", "menu") == null) {
    resetButton(activeButton);    activeButton = null;  }}
function buttonClick(event, menuId) {  var button;
  if  (browser.isIE)        button  =  window.event.srcElement;    else       button  =  event.currentTarget;
button.blur();
  if    (button.menu    ==    null)    {              button.menu    =    document.getElementById(menuId);
menuInit(button.menu);  }  if (activeButton != null)    resetButton(activeButton);
  if (button != activeButton) {    depressButton(button);    activeButton = button;
  }  else    activeButton = null;  return false;}
function buttonMouseover(event, menuId) {  var button;  if (browser.isIE)
    button = window.event.srcElement;  else    button = event.currentTarget;
  if (activeButton != null && activeButton != button)    buttonClick(event, menuId);}
```

```
function depressButton(button) {   var x, y;   button.className += " menuButtonActive";   x = getPageOffsetLeft(button);  y = getPageOffsetTop(button) + button.offsetHeight;  if (browser.isIE) {   x += button.offsetParent.clientLeft;
    y += button.offsetParent.clientTop;  }
  button.menu.style.left = x + "px";   button.menu.style.top  = y + "px";  button.menu.style.visibility = "visible";}
function resetButton(button) {  removeClassName(button, "menuButtonActive");
  if (button.menu != null) {   closeSubMenu(button.menu);
    button.menu.style.visibility = "hidden";  }}
function menuMouseover(event) {    var menu;    if (browser.isIE)        menu = getContainerWith(window.event.srcElement, "DIV", "menu");  else
    menu = event.currentTarget;
  if (menu.activeItem != null)   closeSubMenu(menu);}
function menuItemMouseover(event, menuId) {   var item, menu, x, y;   if (browser.isIE)     item = getContainerWith(window.event.srcElement, "A", "menuItem");   else     item = event.currentTarget; menu = getContainerWith(item, "DIV", "menu");  if (menu.activeItem != null)    closeSubMenu(menu); menu.activeItem = item;  item.className += " menuItemHighlight";
  if (item.subMenu == null) {        item.subMenu = document.getElementById(menuId); menuInit(item.subMenu);    }    x = getPageOffsetLeft(item) + item.offsetWidth;    y = getPageOffsetTop(item);  var maxX, maxY;
  if (browser.isNS) {    maxX = window.scrollX + window.innerWidth;    maxY = window.scrollY + window.innerHeight;  }
  if (browser.isIE && browser.version < 6) {
    maxX = document.body.scrollLeft + document.body.clientWidth;
    maxY = document.body.scrollTop  + document.body.clientHeight;
  }  if (browser.isIE && browser.version >= 6) {    maxX = document.documentElement.scrollLeft + document.documentElement.clientWidth;
    maxY = document.documentElement.scrollTop  + document.documentElement.clientHeight;  }  maxX -= item.subMenu.offsetWidth;
  maxY -= item.subMenu.offsetHeight;
  if (x > maxX)    x = Math.max(0, x - item.offsetWidth - item.subMenu.offsetWidth
      + (menu.offsetWidth - item.offsetWidth));  y = Math.max(0, Math.min(y, maxY));
  item.subMenu.style.left = x + "px";  item.subMenu.style.top  = y + "px";  item.subMenu.style.visibility = "visible";  if (browser.isIE)
    window.event.cancelBubble = true;  else    event.stopPropagation();
}function closeSubMenu(menu) {  if (menu == null || menu.activeItem == null)
    return;    if (menu.activeItem.subMenu != null) {      closeSubMenu(menu.activeItem.subMenu); menu.activeItem.subMenu.style.visibility = "hidden";      menu.activeItem.subMenu = null;  } removeClassName(menu.activeItem, "menuItemHighlight");     menu.activeItem = null;}function menuInit(menu) {
  var itemList, spanList  var textEl, arrowEl;  var itemWidth;  var w, dw;  var i, j;
  if (browser.isIE) {          menu.style.lineHeight = "2.5ex";          spanList = menu.getElementsByTagName("SPAN");   for (i = 0; i < spanList.length; i++)
      if (hasClassName(spanList[i], "menuItemArrow")) {
        spanList[i].style.fontFamily = "Webdings";     spanList[i].firstChild.nodeValue = "4";    }  }
  itemList = menu.getElementsByTagName("A");    if (itemList.length > 0)      itemWidth = itemList[0].offsetWidth;  else   return;
  for (i = 0; i < itemList.length; i++) {   spanList = itemList[i].getElementsByTagName("SPAN")   textEl = null   arrowEl = null;
    for (j = 0; j < spanList.length; j++) {    if (hasClassName(spanList[j], "menuItemText"))     textEl = spanList[j];    if (hasClassName(spanList[j], "menuItemArrow"))     arrowEl = spanList[j];   }   if (textEl != null && arrowEl != null)     textEl.style.paddingRight = (itemWidth      - (textEl.offsetWidth + arrowEl.offsetWidth)) + "px";  }
  if (browser.isIE) {   w = itemList[0].offsetWidth;   itemList[0].style.width = w + "px";
    dw = itemList[0].offsetWidth - w;   w -= dw;   itemList[0].style.width = w + "px";
  }}function getContainerWith(node, tagName, className) {  while (node != null) {
    if (node.tagName != null && node.tagName == tagName &&     hasClassName(node, className)) return node;   node = node.parentNode;
  }  return node;}function hasClassName(el, name) {  var i, list;
```

```
  list = el.className.split(" ");  for (i = 0; i < list.length; i++)    if (list[i] == name)
      return true;  return false;}
function removeClassName(el, name) {   var i, curList, newList;   if (el.className == null)      return;
newList = new Array();  curList = el.className.split(" ");  for (i = 0; i < curList.length; i++)    if (curList[i]
!= name)      newList.push(curList[i]);  el.className = newList.join(" ");}function getPageOffsetLeft(el) {
var x;  x = el.offsetLeft;
  if (el.offsetParent != null)    x += getPageOffsetLeft(el.offsetParent);
  return x;}
function getPageOffsetTop(el) {  var y;  y = el.offsetTop;
  if (el.offsetParent != null)    y += getPageOffsetTop(el.offsetParent);
  return   y;}</script></head><body><!--   Menu   bar.   --><div   class="menuBar"   width="700"><a
class="menuButton"   href="predeterminada.jsp"   onClick="return   buttonClick(event,   'PlantillaMenu');"
onMouseOver="buttonMouseover(event,      'PlantillaMenu');">Plantillas</a><a      class="menuButton"
href="estilopropio.do"            onClick="return            buttonClick(event,            'PersonalizaMenu');"
onMouseOver="buttonMouseover(event, 'PersonalizarMenu');">Personalizar</a>
<a   class="menuButton"   href="Bienvenida.jsp"   onClick="return   buttonClick(event,   'PrimerMenu');"
onMouseOver="buttonMouseover(event,      'PrimerMenu');">Bienvenida</a><a      class="menuButton"
href=""   onClick="return   buttonClick(event,   'helpMenu');"   onMouseOver="buttonMouseover(event,
'helpMenu');">       Help      </a></div><!-- Main menus. -->
<div        id="helpMenu"        class="menu"><a        class="menuItem"        href="menuAyuda.jsp">
Index                     </a></div><script type="text/javascript"></script></body></html>
```

BIENVENIDA.JSP

Descripción: Pantalla de recibimiento al sistema visualizando el usuario. Opciones de Apariencia lista para ser invocada.

Enlaces: Template1, ValidaUsuario, struts-config.xml

Librerias: Taglib de Bean

```
…
<%@include file="/template1.jsp"%>
<!--  beanName="templateLogin.default"  -->
<tiles:insert beanName="template.user" beanScope="request">
<tiles:put name="body-content" type="string" >
 <jsp:include page="Barra.jsp"/>
 <table class="apariencia" cellspacing="2" cellpadding="3" border="0">
 <tr>  <td> <h2 class="title"> Bienvenido  
             <bean:message key="prompt.persona.usuario"/>
               <%=unaPersona.getUsuario()%></h2>
…</tiles:put></tiles:insert>
```

DEFAULTLAYOUT.JSP

Descripción: Define el área de trabajo donde van a estar el header, menú, content-body y copyright.

Enlaces: Stylo.jsp, Bean de Persona

Librerias: Taglib de Tiles, Html

Si el usuario no ha creado una plantilla personalizada, se asignará la plantilla con el StyloPrincipal (Estandar), sino asigna la plantilla con el estilo que selecciónó anteriormente.

```
((unaPersona.getCodigoPlantilla()<=0)||(session.getAttribute("banderaCrear")=="1")){  %>  <jsp:include
page="/StyloPrincipal.jsp"/> <%}else{%>

<jsp:include page="/Stylo1.jsp"/>   <%}%></head>   <body id="body">        <div id="page">   <!--<div
id="content">-->
<table class="apariencia" width="100%" border="0" cellspacing="0" cellpadding="0"><tr>            <!--
Header page information -->
<td colspan="2" height="188">
```

Espacio para la cabecera

```
<tiles:insert attribute="header"/></td>
</tr>      <tr>    <td colspan="2"> <div id="content">
<table    align="left"    width="98%"    height="301"    border="0"    cellpadding="3"    cellspacing="2"
class="apariencia">     <tr>  <!-- Menu bar -->                            <td align="center" valign="top"
width="30%" height="295">
```

Espacio para el Menú vertical

```
<tiles:insert attribute="menubar"/>
</td>     <!-- Main body information -->
<td  width="70%" valign="top" height="295">  <div id="formulario">
```

Espacio para el contenido

```
<tiles:insert attribute="body-content"/>  </div></td>  </tr>
</table>  </div>  </td>  </tr>  <tr>  <!-- Copyright information -->
<td colspan="2">
```

Espacio para el pie de página

```
<tiles:insert attribute="copyright"/>
 </td>     </tr>
</table>

          </div>  <!--</div>--> </body></html>
```

ESTILOPROPIO.JSP

Descripción: Contiene todas las opciones para realizar el cambio de manera personalizada. Este cambio incluye tamaño de fuente, color, tipo de fuente, imagen de cabecera, color de fondo.

Enlaces: Use Bean de Persona, Template1, ValidaUsuario, struts-config.xml, Bean de datos de: TipoFuente, Color Fuente, TamanioFuente, ColorFondo, BuscarObjeto, TipoFuenteEtiqueta, ColorFuenteEtiqueta, TamanioFuenteEtiqueta, TipoFuenteTitulo, TamanioFuenteTitulo, ColorFuenteTitulo, TipoFuenteSubtitulo, TamanioFuenteSubtitulo, ColorFuenteSubtitulo, ColorFondoFormulario, ColorFondoPagina.

Librerias: Taglib de Tags – Bean – Html

Bean que sirve para saber si el usuario ya tiene una plantilla o va a crear una nueva

```
<jsp:useBean id="banderaCrear" class="java.lang.String" scope ="session"/>
```

Bean que sirve para saber que usuario ingresóa la sesión

```
<jsp:useBean id="unaPersona" class="com.bvg.bean.Persona" scope ="session"/>
```

Inclusión del contenedor de la página template1

```
<%@include file="/template1.jsp"%><!--  beanName="templateLogin.default"  -->
<tiles:insert beanName="template.user" beanScope="request">
<tiles:put name="body-content" type="string" >
```

Codificación java que sirve para recibir los BEAN de Datos que se requiere de la BASE de DATOS

```
<%com.bvg.bo.PlantillaBO unaPlantillaBO = new com.bvg.bo.PlantillaBO();        com.bvg.bean.Plantilla
unaPlantillaRecibida = unaPlantillaBO.recuperaCodigo(unaPersona.getCodigoPlantilla());
pageContext.setAttribute("unaPlantillaRecibida",unaPlantillaRecibida);
   System.out.println(unaPlantillaRecibida.getColorFuenteEtiqueta());
   com.bvg.bo.TipoFuenteBO unTipoFuenteBO = new com.bvg.bo.TipoFuenteBO();
   java.util.Vector unalistaTipoFuente = unTipoFuenteBO.buscarTodos();
   pageContext.setAttribute("unalistaTipoFuente",unalistaTipoFuente);
   com.bvg.bo.ColorFuenteBO unColorFuenteBO = new     com.bvg.bo.ColorFuenteBO();java.util.Vector
listaColorFuente = unColorFuenteBO.buscarTodos();
pageContext.setAttribute("listaColorFuente",listaColorFuente);
```

```
com.bvg.bo.TamanioFuenteBO unTamanioFuenteBO = new com.bvg.bo.TamanioFuenteBO();java.util.Vector listaTamanioFuente = unTamanioFuenteBO.buscarTodos();
pageContext.setAttribute("listaTamanioFuente",listaTamanioFuente);
com.bvg.bo.ColorFondoBO unColorFondoBO = new com.bvg.bo.ColorFondoBO();java.util.Vector listaColorFondo = unColorFondoBO.buscarTodos();
pageContext.setAttribute("listaColorFondo",listaColorFondo);%><jsp:include page="/Barra.jsp"/><div id="formulario"><h5 class="subtitle">Usuario <%=unaPersona.getUsuario()%> puedes personalizar el estilo de presentaci&oacute;n del WebSide. </h5> <p><strong>Instrucciones:</strong></p>
<ul><li> Escoge los tipos de fuente de cada objeto</li> <li> Escoge el color que deseas para los objetos Etiquetas, Titulos y Subtitulos</li> <li> Le recomendamos que seleccione<big><font color="#FF0000" > strong>TODAS</strong></font></big> la opciones con los datos a su gusto para ver un efecto agradable,o utilice la ayuda para ver los ejemplos </li> </ul> <% String action;
if (request.getAttribute("banderaCrear")=="1") action = "/crearPlantilla.do";
else action ="/actualizarPlantilla.do"; %>
<html:form action= "<%=action%>" method="post" enctype='multipart/form-data' > <div align="center"> <table class="apariencia" cellspacing="2" cellpadding="3" width="58%"> <tr align="center"> <th align="center" colspan="4">
<div align="center"><label class="title"> Secci&oacute;nFormulario </label></div> </th>
</tr> <tr> <th width="25%"> <div align="right">
<u> <label class="etiqueta">Objetos</label> </u> </div> </th> <td width="25%"> <div align="left">
<label class="etiqueta">Tipo Fuente</label> </div> </td> <td width="19%"> <label class="etiqueta"> Color Fuente </label></td>
<td width="31%"> <label class="etiqueta"> Tama&ntilde;o Fuente </label> </td>
</tr> <tr> <th width="25%"> <div align="right"> <label class="etiqueta"> Etiqueta</label> </div>
</th> <td width="25%"> <html:select styleClass="select" property="tipoFuenteEtiqueta"
value="<%=unaPlantillaRecibida.getTipoFuenteEtiqueta()%>"><html:options
collection="unalistaTipoFuente"labelProperty="tipoFuente" property="codigoTipoFuente"/>
</html:select> </td> <td width="19%">
<html:select styleClass="select" property="colorFuenteEtiqueta"
```

Selecciona en el combo el valor del atributo del objeto que se ha creado anteriormente, y de esta manera el usuario pueda modificar solo el atributo que desee volver a cambiar hasta que la personalización quede a su completo gusto.

```
value="<%=unaPlantillaRecibida.getColorFuenteEtiqueta()%>"> <html:options collection="listaColorFuente"
labelProperty="nombreColor" property="codigocolorFuente"/> </html:select> </td><td width="31%"><html:select styleClass="select" property="tamanioFuenteEtiqueta"
value="<%=unaPlantillaRecibida.getTamanioFuenteEtiqueta()%>">
<html:options collection="listaTamanioFuente"labelProperty="tamnioFuente" property="codigoTamanioFuente"/></html:select> </td> </tr> <tr> <th width="25%"> <div align="right"><label class="etiqueta">Titulos </label> </div> </th><td width="25%"><html:select styleClass="select" property="tipoFuenteTitulo"
value="<%=unaPlantillaRecibida.getTipoFuenteTitulo()%>"><html:options
collection="unalistaTipoFuente"labelProperty="tipoFuente"
property="codigoTipoFuente"/></html:select> </td> <td width="19%"><html:select styleClass="select"property="colorFuenteTitulo"
value="<%=unaPlantillaRecibida.getColorFuenteTitulo()%>">
<html:options collection="listaColorFuente"
labelProperty="nombreColor" property="codigocolorFuente"/></html:select></td><td width="31%"><html:select styleClass="select" property="tamanioFuenteTitulo"
value="<%=unaPlantillaRecibida.getTamanioFuenteTitulo()%>"> <html:options collection="listaTamanioFuente" labelProperty="tamnioFuente"
property="codigoTamanioFuente"/></html:select></td> </tr> <tr> <th width="25%"><div align="right"><label class="etiqueta">Subtitulos</label></div></th><td><html:select styleClass="select" property="tipoFuenteSubtitulo"value="<%=unaPlantillaRecibida.getTipoFuenteSubtitulo()%>">
<html:options collection="unalistaTipoFuente" labelProperty="tipoFuente"
property="codigoTipoFuente"/> </html:select></td><td><html:select styleClass="select"property="colorFuenteSubtitulo"
```

```
value="<%=unaPlantillaRecibida.getColorFuenteSubtitulo()%>">                    <html:options
collection="listaColorFuente"          labelProperty="nombreColor"property="codigocolorFuente"/>
</html:select></td><td><html:select       styleClass="select"     property="tamanioFuenteSubtitulo"
value="<%=unaPlantillaRecibida.getTamanioFuenteSubtitulo()%>"><html:options
collection="listaTamanioFuente"labelProperty="tamnioFuente"      property="codigoTamanioFuente"/>
</html:select></td> </tr> <tr> <th > </th>
<td width="25%">  <div align="left"> <label class="etiqueta">Tipo Fuente</label>
</div>  </td>  <td  width="19%">  <label  class="etiqueta">  Color  Fuente  </label></td>          <td
width="31%"> <label class="etiqueta"> Color Fondo </label> </td>   </tr>
<tr>  <th> <div align="right"><label class="etiqueta">Listas (list)</label> </div></th>
  <td><html:select styleClass="select" property="tipoFuenteLista"
  value="<%=unaPlantillaRecibida.getTipoFuenteLista()%>">
  <html:options          collection="unalistaTipoFuente"               labelProperty="tipoFuente"
property="codigoTipoFuente"/></html:select></td><td><html:selectstyleClass="select"
property="colorFuenteLista"
  value="<%=unaPlantillaRecibida.getColorFuenteLista()%>"><html:options
collection="listaColorFuente"labelProperty="nombreColor"
property="codigocolorFuente"/></html:select></td><td><html:select              styleClass="select"
property="colorFondoLista"
  value="<%=unaPlantillaRecibida.getColorFondoLista()%>">
  <html:options                         collection="listaColorFondo"labelProperty="descripcion"
property="codigoColorFondo"/></html:select></td>      </tr>      <tr><th><div     align="right"><label
class="etiqueta"> Combos (select)</label></div> </th>
<td><html:select            styleClass="select"        property="tipoFuenteCombo"><html:options
collection="unalistaTipoFuente"                                       labelProperty="tipoFuente"
property="codigoTipoFuente"/></html:select></td>   <td><html:select              styleClass="select"
property="colorFuenteCombo"
value="<%=unaPlantillaRecibida.getColorFuenteCombo()%>"><html:options
collection="listaColorFuente"labelProperty="nombreColor"
property="codigocolorFuente"/></html:select></td>   <td><html:select
styleClass="select"property="colorFondoCombo"
value="<%=unaPlantillaRecibida.getColorFondoCombo()%>">
<html:options collection="listaColorFondo"
labelProperty="descripcion"  property="codigoColorFondo"/></html:select></td>  </tr>     <tr><th><div
align="right"><label class="etiqueta">Boton</label></div> </th>
<td><html:select styleClass="select" property="tipoFuenteBoton"
value="<%=unaPlantillaRecibida.getTipoFuenteBoton()%>"><html:options
collection="unalistaTipoFuente"labelProperty="tipoFuente"
property="codigoTipoFuente"/></html:select></td><td><html:select              styleClass="select"
property="colorFuenteBoton"
value="<%=unaPlantillaRecibida.getColorFuenteBoton()%>"><html:options
collection="listaColorFuente"labelProperty="nombreColor"
property="codigocolorFuente"/>/html:select></td><td><html:select
styleClass="select"property="colorFondoBoton"
value="<%=unaPlantillaRecibida.getColorFondoBoton()%>"><html:options
collection="listaColorFondo"labelProperty="descripcion"
property="codigoColorFondo"/> </html:select></td> </tr>  <tr>  <th width="25%">
<div    align="right"><label    class="etiqueta">    Color    de    Fondo</label>    </div></th><td
colspan="4"><html:select          styleClass="select"          property="colorFondoFormulario"
value="<%=unaPlantillaRecibida.getColorFondoFormulario()%>"><html:options
collection="listaColorFondo"                                    labelProperty="descripcion"
property="codigoColorFondo"/></html:select></td></tr></table><table  class="apariencia"  width="398">
<tr>   <th width="31%" colspan="4">
<div align="center" class="title"> Secci&oacute;n Pagina</div>        </th>
</tr><tr><th><div align="right"><label class="etiqueta"> Objetos</label></div></th>
 <td><label class="etiqueta"> Color</label></td> </tr> <tr>  <th width="31%">
 <div  align="right"><label  class="etiqueta">  Color  de  Fondo</label>  </div>           </th><td
width="30%"><html:select              styleClass="select"          property="colorFondoPagina"
```

```
value="<%=unaPlantillaRecibida.getColorFondoPagina()%>"> <html:options
collection="listaColorFondo"
labelProperty="descripcion" property="codigoColorFondo"/>
</html:select></td> </tr> <tr> <th width="31%"><div align="right"><label class="etiqueta"> Color de
Barra Menu;</label> </div> </th><td width="30%"><html:select styleClass="select"
property="colorBorde"> <html:options collection="listaColorFondo"
labelProperty="descripcion"property="codigoColorFondo"/></html:select> </td> <tr> <th>
<div align="right"> <label class="etiqueta"> Imagen de Cabecera </label> </div></th> <td
width="30%"><html:file property="imagenCabecera" onchange="cargar(1);"/> </td> </tr> </table>
</div><div align="center"><input type="submit" value="Crear Estilo" class="submit"/><html:hidden
property="usuario"
value="<%=unaPersona.getUsuario()%>"/><html:hidden property="codigoPlantilla"
value="<%=unaPersona.getCodigoPlantilla()%>"/></div></html:form></div> </tiles:put></tiles:insert>
```

PREDETERMINADA.JSP

Descripción: Muestra al usuario en forma amigable con una ventana de imágenes en miniatura de las plantillas que puede seleccionar para cambiar su apariencia.

Enlaces: Template1, ValidaUsuario, struts-config.xml, BuscarPlantilla

Librerias: Taglib de Tags – Bean – Html - Tiles

Imágenes: Archivos Gifs

```
Cambio de estilo
<script language="javascript" type="text/javascript">

Retardo entre imagenes en la reproduccion automatica (milisegundos)
var retardo_rotacion = 1000
//indice imagen inicial
actual = 0

Ir a la siguiente imagen
function siguiente() {
  if (document.forms[0].codigoPlantilla[actual+1]) {
    document.images.visor.src = document.forms[0].codigoPlantilla[actual+1].value
    document.forms[0].codigoPlantilla.selectedIndex = ++actual
  }
  else ir_inicio()
}

Ir a la imagen anterior
function anterior() {
  if (actual-1 >= 0) {
    document.images.visor.src = document.forms[0].codigoPlantilla[actual-1].value
    document.forms[0].codigoPlantilla.selectedIndex = --actual
```

```
      }
   else ultimo()
}
```

Ir a la primera imagen

```
function ir_inicio() {
   actual = 0
   document.images.visor.src = "estilos/Plantilla.gif"
   document.forms[0].codigoPlantilla.selectedIndex = 0
}
```

Ir a la ultima imagen

```
function ultimo() {
   actual = document.forms[0].codigoPlantilla.length-1
   document.images.visor.src = document.forms[0].codigoPlantilla[actual].value
   if(document.forms[0].codigoPlantilla[actual].value==1){
   document.images.visor.src = "estilos/plantilla.gif"
   }  if(document.forms[0].codigoPlantilla[actual].value==2){
   document.images.visor.src = "estilos/plantilla2.gif"
   }  if(document.forms[0].codigoPlantilla[actual].value==3){
   document.images.visor.src = "estilos/plantillaA.gif"
   }  if(document.forms[0].codigoPlantilla[actual].value==4){
       document.images.visor.src = "estilos/plantilla3.gif"
    }  if(document.forms[0].codigoPlantilla[actual].value==5){
  document.images.visor.src = "estilos/plantilla4.gif"    }
   document.forms[0].codigoPlantilla.selectedIndex = actual}
```

Cambia a modo reproduccion automatica

```
function repAuto(text) {
   document.forms[0].automatico.value = (text == "Detener") ? "Reproducir" : "Detener"
   reproducir()
}function cambiaLista() {
   actual = document.forms[0].codigoPlantilla.selectedIndex
   if(document.forms[0].codigoPlantilla[actual].value==1){                    document.images.visor.src          =
"estilos/plantilla.gif"   }
   if(document.forms[0].codigoPlantilla[actual].value==2){                      document.images.visor.src        =
"estilos/plantilla2.gif"   }
   if(document.forms[0].codigoPlantilla[actual].value==3){
   document.images.visor.src = "estilos/plantillaA.gif"   }
   if(document.forms[0].codigoPlantilla[actual].value==4){
       document.images.visor.src = "estilos/plantilla3.gif"       }
  if(document.forms[0].codigoPlantilla[actual].value==5){
  document.images.visor.src = "estilos/plantilla4.gif"       }}
```

Realiza reproducción atuomática

```
function reproducir() {
   if (document.forms[0].automatico.value == "Detener") {
      actual = (actual == document.forms[0].codigoPlantilla.length-1) ? 0 : actual+1
      if(document.forms[0].codigoPlantilla[actual].value==1){
        document.images.visor.src = "estilos/plantilla.gif"        }
      if(document.forms[0].codigoPlantilla[actual].value==2){
        document.images.visor.src = "estilos/plantilla2.gif"        }
       if(document.forms[0].codigoPlantilla[actual].value==3){
       document.images.visor.src = "estilos/plantillaA.gif"       }
       if(document.forms[0].codigoPlantilla[actual].value==4){
       document.images.visor.src = "estilos/plantilla3.gif"       }
       if(document.forms[0].codigoPlantilla[actual].value==5){
       document.images.visor.src = "estilos/plantilla4.gif"       }
      document.forms[0].codigoPlantilla.selectedIndex = actual
```

```
window.setTimeout("reproducir()", retardo_rotacion)  }}</script>
```

Sirve para traer de la base de datos el código y el nombre de la plantilla que se van a mostrar en el select CodigoPlantilla

```
<%  com.bvg.bo.PlantillaBO unaPlantillaDAO = new com.bvg.bo.PlantillaBO();
 java.util.Vector unaListaPlnatilla = unaPlantillaDAO.buscarTodos();
pageContext.setAttribute("unalistaPlantilla",unaListaPlnatilla);
String codigoPlantilla = (String)request.getAttribute("codigoPlantilla");
if((codigoPlantilla==null)||(codigoPlantilla.equalsIgnoreCase("null")) || (codigoPlantilla.trim().equals(""))){
codigoPlantilla="01";com.bvg.bo.PlantillaBO  unaPlantillaBO  =  new  com.bvg.bo.PlantillaBO();    }%>
<jsp:include page="/Barra.jsp"/><div id="formulario">
 <p id="title">Bienvenido <%=unaPersona.getUsuario()%></p>
 …<html:select styleClass="select" property="codigoPlantilla"
onchange="cambiaLista();">
<html:options collection="unalistaPlantilla"
labelProperty="nombre" property="codigoPlantilla"/>
```

Combo que contiene el código y el nombre de la plantilla

```
</html:select>   </td>    </tr>   <tr> <td align="center" bgcolor="#C0C0C0">
```

Inserción de botones inicio, anterior, automático, reproducción automática, etc.

```
<input type="button" style="submit" onClick="ir_inicio();" value="|<<" title="Inicio">
...
</html:form>
</tiles:put>
</tiles:insert>
```

STYLO1.JSP

Descripción: Frame que contiene la etiqueta <style> </style> que utilizamos para crear el estilo en forma dinámica.

Enlaces: Template1, ValidaUsuario, struts-config.xmlm, BuscarPlantilla

Librerias: Taglib de Tags – Bean – Html - Tiles

Imágenes: Archivos Gifs

Recibir los datos de la Plantilla que estaba guardada en la base de datos
Buscarplantilla es el método de la clases PlantillaDAO que permite obtener todos los atributos que necesitamos de la base de datos

```
int codigoPlantilla = unaPersona.getCodigoPlantilla();
com.bvg.bean.Plantilla unaPlantilla = new com.bvg.bean.Plantilla();
com.bvg.bo.PlantillaBO unaPlantillaBO = new com.bvg.bo.PlantillaBO();
unaPlantilla = unaPlantillaBO.buscarPlantilla(codigoPlantilla);
pageContext.setAttribute("unaPlantilla",unaPlantilla);
session.setAttribute("unaPlantilla",unaPlantilla);
```

El método escribir de la clase escribir Imagen permite bajar el archivo tipo byte de la tabla imagen de la Base de Datos para mostrarlo en la cabecera, puede ser cualquier archivo de imagen

```
com.bvg.util.EscribirImagen ecrituraImagen = new com.bvg.util.EscribirImagen();
String rutafoto= ecrituraImagen.escribir(unaPersona.getUsuario(),unaPlantilla.getImagenFoto());
  pageContext.setAttribute("rutafoto",rutafoto);%>
```

Etiqueta de estilos en cascada

```
<style type="text/css" media="screen, print" >
```

Atributos del cuerpo de la página, con un color de fondo elegido por el usuario.

```
body {    background-color: <%=unaPlantilla.getColorFondoPagina()%>;
          font-family: Arial, Helvetica, sans-serif;
          font-size: 12px;line-height: 24px; color: #333333;}

#page {width:900px;     margin:0 0;     height:100%;}
```

Atributos de la cabecera de la página, que tiene un background personalizado con información de la base de datos

```
#header {width:1000px;height:150px;margin: 0 0;    background:
url('<%=com.bvg.util.Parametro.ruta%><%=unaPlantilla.getImagenHeader()%>')     no-repeat;
          padding:10px 20px 10px 20px;}
```

Atributos del contenido de la página, que tiene un background personalizado con información de la base de datos

```
#content {padding:0px 20px 0 20px;
background:url('<%=com.bvg.util.Parametro.ruta%><%=unaPlantilla.getImagenBody()%>')    repeat-
y;height:100%;   width:1000px;margin: 0 auto;   background-position:inherit;}
```

Atributos del formulario de la página, que tiene un background con un color personalizado con información de la base de datos

```
#formulario {
padding:0px 20px 0 20px;
background:<%=unaPlantilla.getColorFondoFormulario()%>height:100%;    width:700px;   margin:  0px
40px;   background-position:inherit;padding:0px 20px 20px 20px;}
```

Atributos de las tablas que se utilizan en la etiqueta display de las páginas.

```
table {    border: 0px solid #666;width: 80%;
       font-family: <%=unaPlantilla.getTipoFuenteEtiqueta()%>, Helvetica, sans-serif;
}th, td {   vertical-align: top;}thead tr {  background-color: #fc0;}td.hit {
    vertical-align: top;    padding: 2px 4px 2px 4px;*/    text-align: left;
    font-family: <%=unaPlantilla.getTipoFuenteEtiqueta()%>, Helvetica, sans-serif;
    font-size: 11px;    text-decoration: underline;    color: black;}
th.sorted {    background-color: orange;}th a, th a:visited {  color: black;}
th a:hover {  text-decoration: underline;  color: black;}th.sorted a, th.sortable a {
background-position: right;display: block;     width: 100%;}
th.sortable a {      background-image: url(../img/arrow_off.png);}th.order1 a {
```

background-image: url(../img/arrow_down.png);}th.order2 a { background-image: url(../img/arrow_up.png);}tr.odd { background-color: #fff}tr.tableRowEven, tr.even { background-color: #fea}
ul#showsource {float: right; background-color: #eee; border: 1px dotted #999; padding: 2px 4px 2px 4px; margin: 5px; width:100px; list-style-type: none;}

Estilos de tabla para el display
table.its thead tr { background-color: #69c;}table.its tr.even { background-color: #def;}

table.mars thead tr { background-color: #9c9;}table.mars tr.even { background-color: #ccb;}table.mars tr.odd { background-color: #eec;}table.simple thead tr {
background-color: #eee;}table.simple tr.even { background-color: #fff;}
table.simple { border: 1px solid #ccc; border-collapse: collapse;}table.simple td,table.simple th{border: 1px solid #ccc;} table.buscador{border: 0px solid #666;width: 80%;margin: 20px 4px 20px 4px;}table.buscador tr.even { background-color: #fff;}
table.buscador td.hit,{ border: 0px solid #ccc;}
table.nocol tbody td,table.nocol tbody th{ border-left: none;border-right: none;
table.report td, table.report th{font: menu;}
table.report th{ background: buttonface; border-width: 1px;border-style: solid;
border-color: threedhighlight threedshadow threedshadow threedhighlight;cursor: hand;}
table.report idcol {background: buttonface;}
table.report tr.even{background-color: #399;}
table.report tbody {height: 100px;overflow: auto;}
table.report { height: 120px; overflow: auto;}
table.report tbody tr{ height: 10px;}
table.mark td.tableCellError { background-color: #d00;color: #fff;}
table.sublist { margin: 0 0 0 0; width: 100%;}
table.sublist td, table.sublist th {width: 50%;}
table.apariencia { border: 0px;width: 80%; margin: 0 0 0 0;}
div.exportlinks { background-color: #eee; border: 1px dotted #999; padding: 2px 4px 2px 4px;margin: -10px 0 10px 0;width: 79%;}
span.export { padding: 0 4px 1px 20px; display:inline; display:inline-block;cursor: pointer;}span.excel {background-image: url(../img/ico_file_excel.png);}
span.csv {background-image: url(../img/ico_file_csv.png);}
span.xml {background-image: url(../img/ico_file_xml.png);}
span.pdf {background-image: url(../img/ico_file_pdf.png);}
span.pagebanner {background-color: #eee; border: 1px dotted #999; padding: 2px 4px 2px 4px; width: 79%;margin-top: 10px;display:block; border-bottom: none;}span.pagelinks {background-color: #eee; border: 1px dotted #999;padding: 2px 4px 2px 4px;width: 79%;display:block; border-top: none;margin-bottom: -5px;}
table.simple td.textRed {color: #b11;}
fin de estilos de tabla para los display

Dar el color de fondo a la etiqueta Form
form { <%=unaPlantilla.getColorFondoFormulario()%>;}

Atributos de las etiquetas para el estilo de los títulos
h1{color:#fff111;font-weight:normal;font-size:24px;padding: 40px 0 10px 40px;}

h2 a{ position:absolute;top:32px;color:#666;font-size:16px;margin:0 0 0 575px; text-decoration:none;}

h2 a:hover{ color:#999;}color:black;font-size:12px;font-weight:bold; padding: 2px 0 2px 2px

h4{color:#8d8d8d;font-size:20px;font-weight:normal;}

h5{color:#8d8d8d;font-size:10px;font-weight:normal;padding:5px;}

h3 img{float:left;}h2 u{position:absolute; top:32px;color:#666;font-size:16px;margin:0 0 0 700px;text-decoration:none; font-family: Arial, Helvetica, sans-serif;}

Es una clase de estilo para título personalizado

```
.title {font-family:
<%=unaPlantilla.getTipoFuenteTitulo()%>, "Times New Roman", Times, serif;
font-size: <%=unaPlantilla.getTamanioFuenteTitulo()%>;        line-height: 30px;
background-color: #FFFF66; color: <%=unaPlantilla.getColorFuenteTitulo()%>;}
```

Es una clase de estilo para subtítulo personalizado

```
.subtitle {font-family: <%=unaPlantilla.getTipoFuenteSubtitulo()%>, "Times New Roman", Times, serif;
font-size: <%=unaPlantilla.getTamanioFuenteSubtitulo()%>;
line-height: 20px;font-weight: bold;color:
<%=unaPlantilla.getColorFuenteSubtitulo()%>; font-style: oblique;}
```

Es una clase de estilo para cabecera

```
.header {font-family: Georgia, "Times New Roman", Times, serif;        font-size:        24px;background-
color: #FFFF66;color: #FFFF66;}
```

Etiquetas para menú vertical

```
#nav_top {
background:url('<%=com.bvg.util.Parametro.ruta%>/imageRevival/nav_top.gif')        no-repeat;
        width:155px;height:28px;}
#nav_btm {
background:url('<%=com.bvg.util.Parametro.ruta%>/imageRevival/nav_btm.gif')        no-repeat;
        width:155px;height:28px;}
#menu {float:center;        width: 150px;      color:#000;
background:url('<%=com.bvg.util.Parametro.ruta%>/imageRevival/nav_mid.gif')   repeat-y;   margin:0px
0px 0px 0px;}
#menu ul {list-style:none;width:100px;font-size:12px;padding:0 0 0 3px;}
#menu li {list-style-type: none;        //background:transparent
url('C:/JDEVADF_10.1.3/jdev/mywork/Academico1/ViewController/public_html/Imagenes/imageRevival/
bullet.gif') no-repeat 12px 6px; margin-bottom:4px;border-bottom:1px solid #c8e5c7;padding:0;}
#menu li:hover {background:#c8e5c7;margin-bottom:4px;}
#menu a {display:block;padding:0 0 0 20px; color:#fff;text-decoration: none;background:#f0f6df;}
#menu a:hover {display:block;padding:0 0 0 20px;     color:#333;text-decoration:    none;background:fff;}
        }.
legal {font-family: Georgia, "Times New Roman", Times, serif;font-size: 12pt;color: #333333;}
.box1 {   border-width: 2px;border-color: #CCCCCC #333333 #333333 #CCCCCC;
border-style: dotted;}
.promo {font-family: "Times New Roman", Times, serif;color: #000033;}
.titlebar {font-family: "Times New Roman", Times, serif;font-size: 9px;color: #FFFFFF;background-color:
#336699;}
.dingbat {font-family: Georgia, "Times New Roman", Times, serif;        background-color:     #CCCC99;
color: #660000; font-weight: bolder; font-size: medium;}
input.big {width: 100px;}
input.small {        width: 50px;}
```

Etiquetas para que se vea el menú como estilo navegador

```
div.menuArriba, div menuArriba a.menuNav{
  font-family: <%=unaPlantilla.getTipoFuenteEtiqueta()%>, Arial, sans-serif;
  font-size: 10pt;  font-style:italic;  font-weight:bold;  color: #FFFFFF;}
div.menuArriba{border-color: #FFFFFF #909090 #909090 #6699FF;padding: 0px 0px 2px 2px;     text-
align: left;}div.menuArriba a.menuNav {
background-color: transparent;   color: #000000;  cursor: auto;  left: 0px;  margin: 1px;  padding: 0px
0px 2px 2px;  position: relative;  text-decoration:blink;  top: 0px;
z-index: 100;}
div.menuArriba a.menuNav:hover {  background-color: #99FFFF; color: #000000;}
```

Etiquetas para que se vea el menú horizontal de barra.

```
div.menuBar,div.menuBar a.menuButton,div.menu,div.menuBuscador
div.menu a.menuItem {  font-family: MS Sans Serif, Arial, sans-serif;  font-size: 8pt;  font-style: normal; font-weight: normal;  color: #0099FF;}
div.menuBar { background-color: #6699FF;  border: 2px solid;  border-color: #6699FF #909090 #909090 #6699FF;
  padding: 4px 2px 4px 2px;  text-align: left;}
div.menuBuscador {  background-color: #6699FF;
border: 2px solid;  border-color: #6699FF #909090 #909090 #6699FF;
padding: 4px 2px 4px 2px;  text-align: left;}
div.menuBar a.menuButton {
background-color: transparent;  border: 1px solid #6699FF;  color: #000000;  cursor: auto;  left: 0px; margin: 1px;  padding: 2px 6px 2px 6px;  position: relative;  text-decoration: none;  top: 0px;  z-index: 100;}
div.menuBuscador a.menuButton:hover {  background-color: #FFFFFF;  border-color: #6699FF #909090 #909090 #6699FF;  color: #000000;}
div.menuBuscador a.menuButton {  background-color: transparent;  border: 1px solid #6699FF;  color: #000000;  cursor: auto;  left: 10px;  margin: 1px;
padding: 2px 6px 2px 6px;  position: relative;  text-decoration: none;  top: 0px;  z-index: 100;}
div.menuBar a.menuButton:hover {
 background-color: #FFFFFF;  border-color: #6699FF #909090 #909090 #6699FF;  color: #000000;}
div.menuBar a.menuButtonActive,div.menuBar a.menuButtonActive:hover {
 background-color: #a0a0a0;  border-color: #909090 #f0f0f0 #f0f0f0 #909090;
color: #ffffff;  left: 1px;  top: 0px;}
div.menu {  background-color: #d0d0d0;
 border: 2px solid;  border-color: #f0f0f0 #909090 #909090 #f0f0f0;  left: 0px;  padding: 0px 1px 1px 0px;  position: absolute;  top: auto;  visibility: hidden; z-index: 101;}
div.menu a.menuItem {  color: #000000;  cursor: auto;  display: block;  padding: 3px 1em;  text-decoration: none;  white-space: nowrap;  top:auto;}
div.menu a.menuItem:hover, div.menu a.menuItemHighlight {
 background-color: #000080;  color: #ffffff;}
div.menu a.menuItem span.menuItemText {}div.menu a.menuItem span.menuItemArrow {  margin-right: -.75em;}
div.menu div.menuItemSep {
 border-top: 1px solid #909090;  border-bottom: 1px solid #f0f0f0;  margin: 4px 2px;}
```

Estilos que puede modificar el usuario

```
.etiqueta {font-family: <%=unaPlantilla.getTipoFuenteEtiqueta()%>, Helvetica, sans-serif;color: <%=unaPlantilla.getColorFuenteEtiqueta()%>;  font-size: <%=unaPlantilla.getTamanioFuenteEtiqueta()%>;  font-style:normal;font-weight:bold;
}.submit
{color:<%=unaPlantilla.getColorFondoBoton()%>;font:<%=unaPlantilla.getTipoFuenteBoton()%>, Times New Roman, Times, serif;font-color:<%=unaPlantilla.getColorFuenteBoton()%>;font-size:14px;width:100;caption-side:bottom;}
.select {font-family:
<%=unaPlantilla.getTipoFuenteCombo()%>, Helvetica,
sans-serif;  background-color:
 <%=unaPlantilla.getColorFondoCombo()%>;font-size: 12px;
color: <%=unaPlantilla.getColorFuenteCombo()%>;}
.list {     font-family: <%=unaPlantilla.getTipoFuenteLista()%>,
Helvetica, sans-serif;
 background-color:
<%=unaPlantilla.getColorFondoLista()%>;font-size:
 12px; color: <%=unaPlantilla.getColorFuenteLista()%>;}
</style>
```

1.3. CLASES JAVA

**Bean**

ColorFondo.java

Descripción: Se declara los campos que va a ser usado por el usuario, el codigo del color del fondo y su descripción.

Métodos: Get y Set.

Librerías Utilizadas: com.bvg.bean

```
public class ColorFondo {
private String codigoColorFondo;private String descripcion;
public ColorFondo() { }
    public void setDescripcion(String descripcion) { this.descripcion = descripcion;  }
    public String getDescripcion() {  return descripcion;   }
    public void setCodigoColorFondo(String codigoColorFondo) {
    this.codigoColorFondo = codigoColorFondo;   }
    public String getCodigoColorFondo() {return codigoColorFondo;  }}
```

ColorFuente.java

Descripción: Se declara los campos que va a ser usado por el usuario, el codigo del color de la fuente, nombre del color y su RGB.

Métodos: Get y Set.

Librerías Utilizadas: com.bvg.bean

```
public class Colorfuente {     private String codigocolorFuente;     private String nombreColor;    private
String rgb;   public Colorfuente() {    }
public void setCodigocolorFuente(String codigocolorFuente) {               this.codigocolorFuente =
codigocolorFuente;   }
public String getCodigocolorFuente() {  return codigocolorFuente;   }
public void setNombreColor(String nombreColor) {this.nombreColor = nombreColor;    }public String
getNombreColor() { return nombreColor; }
public void setRgb(String rgb) { this.rgb = rgb;  }public String getRgb() { eturn rgb;}}
```

Etiqueta.java

Descripción: Se declara los campos que va a ser usado por el usuario, el codigo de la etiqueta, codigo del tipo de la fuente, codigo del color de la fuente, codigo del tamaño de la fuente y su estilo.

Métodos: Get y Set.

Librerías Utilizadas: com.bvg.bean

```
public class Etiqueta {   private int codigoEtiqueta;   private String codigoTipoFuente;
private String codigoColorFuente;   private String codigoTamanioFuente;   private String estilo;
public Etiqueta() {   }public void setCodigoColorFuente(String codigoColorFuente) {
this.codigoColorFuente = codigoColorFuente;   }public String getCodigoColorFuente() {return
codigoColorFuente;   }public void setCodigoTamanioFuente(String codigoTamanioFuente)
{this.codigoTamanioFuente = codigoTamanioFuente;}   public String getCodigoTamanioFuente()
{return codigoTamanioFuente;  }public void setEstilo(String estilo) { this.estilo = estilo; } public String
getEstilo() { return estilo;  }
public void setCodigoTipoFuente(String codigoTipoFuente) {   this.codigoTipoFuente =
codigoTipoFuente;   }  public String getCodigoTipoFuente() {return codigoTipoFuente; }  public void
setCodigoEtiqueta(int codigoEtiqueta) {   this.codigoEtiqueta = codigoEtiqueta;   } public int
getCodigoEtiqueta() {return codigoEtiqueta; }}
```

FondoFormulario.java

Descripción: Se declara los campos que va a ser usado por el usuario, el código del fondo del formulario, código del color del formulario y su imagen.

Métodos: Get y Set.

Librerías Utilizadas: com.bvg.bean

```
public class FondoFormulario {   private int codigoFondoFormulario;   private String codigoColorFondo;
private String imagen;   public FondoFormulario() {   }
public void setCodigoFondoFormulario(int codigoFondoformulario) {   this.codigoFondoFormulario =
codigoFondoformulario;  }
public int getCodigoFondoFormulario() { return codigoFondoFormulario; }
public void setCodigoColorFondo(String codigoColorFondo) {   this.codigoColorFondo =
codigoColorFondo;  }
public String getCodigoColorFondo() {return codigoColorFondo;  }
public void setImagen(String imagen) { this.imagen = imagen;  }
public String getImagen() {return imagen;  }}
```

FondoPagina.java

Descripción: Se declara los campos que va a ser usado por el usuario, el codigo del fondo de la pagina , el codigo del color del fondo y el codigo de la imagen.

Métodos: Get y Set.

Librerías Utilizadas: com.bvg.bean

```
public class FondoPagina {  private int codigoFondoPagina; private String codigoColorFondo; private int
codigoImagen;  public FondoPagina() {   }
public void setCodigoFondoPagina(int codigoFondoPagina) {          this.codigoFondoPagina =
codigoFondoPagina;   }
public    int    getCodigoFondoPagina()    {return    codigoFondoPagina;      }          public    void
setCodigoColorFondo(String codigoColorFondo) {this.codigoColorFondo = codigoColorFondo;  }  public
String   getCodigoColorFondo()   {return   codigoColorFondo;   }    public   void   setCodigoImagen(int
codigoImagen) {          this.codigoImagen = codigoImagen;    } public int getCodigoImagen() {return
codigoImagen; }}
```

Imagen.java

Descripción: Se declara los campos que va a ser usado por el usuario, el codigo de la imagen, su descripción y el archivo de la imagen.

Métodos: Get y Set.

Librerías Utilizadas: com.bvg.bean

```
public class Imagen {   private int codigoImagen;    private String descripción;    private byte[] archivo;
public Imagen() {   }
public void setCodigoImagen(int codigoImagen) {this.codigoImagen = codigoImagen;     }   public int
getCodigoImagen()   {return   codigoImagen;   }public   void   setDescripción(String   descripción)   {
this.descripción = descripción; } public String getDescripción() { return descripción;   }     public void
setArchivo(byte[] archivo) {        this.archivo = archivo;   }   public byte[] getArchivo() {    return archivo;
}}
```

Parametros.java

Descripción: Se declara los campos que va a ser usado por el usuario, el codigo del parametro y su dato.

Métodos: Get y Set.

Librerías Utilizadas: com.bvg.bean

```
public class Parametro { public static String ruta =
"C:/jdevestudio1013/jdev/mywork/Academico1/ViewController/public_html/Imagenes";}
```

Persona.java

Descripción: Se declara los campos que va a ser usado por el usuario, su usuario, su clave, el nombre, el apellido, el codigo de la plantilla y su archivo.

Métodos: Get y Set.

package com.bvg.bean;

```
public class Persona {    private String usuario;    private String clave;    private String nombre;    private
String apellido;    private int codigoPlantilla;    private String archivo;    public Persona() {    }    public
void setUsuario(String usuario) {        this.usuario = usuario;    }    public String getUsuario() {        return
usuario;    }
public void setClave(String clave) {    this.clave = clave;  }   public String getClave() {        return clave;  }
public void setNombre(String nombre) {this.nombre = nombre;  }
public String getNombre() {return nombre; }  public void setApellido(String apellido) {
this.apellido = apellido;   } public String getApellido() { return apellido; } public void setArchivo(String
archivo) { this.archivo = archivo;   } public String getArchivo() {          return archivo;   }   public void
setCodigoPlantilla(int codigoPlantilla) {            this.codigoPlantilla = codigoPlantilla;    }    public int
getCodigoPlantilla() {        return codigoPlantilla;   }}
```

Plantilla.java

Descripción: Se declara los campos que va a ser usado por el usuario, el codigo de la plantilla, el nombre de la plantilla, el tipo de fuente de la etiqueta, el color de la fuente de la etiqueta, el tamaño de la fuente de la etiqueta, el

estilo de la fuente de la etiqueta, el tipo de fuente del titulo, el color de la fuente del titulo, el tamaño de la fuente del titulo, el estilo de la fuente del titulo, el tipo de la fuente del subtitulo, el color de la fuente del subtitulo, el tamaño de la fuente del subtitulo, el estilo de la fuente del subtitulo, el color del fondo de pagina, el color del fondo del formulario y su imagen de fondo.

Métodos: Get y Set.

Librerías Utilizadas: com.bvg.bean

```
public class Plantilla {private int codigoPlantilla; private String nombrePlantilla; private String
tipoFuenteEtiqueta; private String colorFuenteEtiqueta;private String tamanioFuenteEtiqueta; private
String estiloFuenteEtiqueta;private String tipoFuenteTitulo; private String colorFuenteTitulo;private
String tamanioFuenteTitulo;
private String estiloFuenteTitulo;private String tipoFuenteSubtitulo; private String colorFuenteSubtitulo;
private String tamanioFuenteSubtitulo;private String estiloFuenteSubtitulo; private String
colorFondoPagina;private String colorFondoFormulario; private int ImagenFoto;private String
imagenFondo; private String imagenHeader; private String imagenBody; private String imagenFooter;
private String tipoFuenteCombo; private String colorFuenteCombo; private String colorFondoCombo;
private String tipoFuenteLista; private String colorFuenteLista; private String colorFondoLista; private
String tipoFuenteBoton;private String colorFuenteBoton; private String colorFondoBoton; public
Plantilla() { } public void setCodigoPlantilla(int codigoPlantilla) { this.codigoPlantilla = codigoPlantilla; }
public int getCodigoPlantilla() { return codigoPlantilla; } public void setNombrePlantilla(String
nombrePlantilla) { this.nombrePlantilla = nombrePlantilla; } public String getNombrePlantilla() { return
nombrePlantilla; }public void setTipoFuenteEtiqueta(String tipoFuenteEtiqueta) {this.tipoFuenteEtiqueta
= tipoFuenteEtiqueta; } public String getTipoFuenteEtiqueta() {return tipoFuenteEtiqueta; } public void
setColorFuenteEtiqueta(String colorFuenteEtiqueta) { this.colorFuenteEtiqueta = colorFuenteEtiqueta;
}public String getColorFuenteEtiqueta() {return colorFuenteEtiqueta; }public void
setTamanioFuenteEtiqueta(String tamanioFuenteEtiqueta) { this.tamanioFuenteEtiqueta =
tamanioFuenteEtiqueta; }public String getTamanioFuenteEtiqueta() { return
tamanioFuenteEtiqueta;}public void setEstiloFuenteEtiqueta(String estiloFuenteEtiqueta)
{this.estiloFuenteEtiqueta = estiloFuenteEtiqueta; } public String getEstiloFuenteEtiqueta() {return
estiloFuenteEtiqueta; } public void setTipoFuenteTitulo(String tipoFuenteTitulo) {
this.tipoFuenteTitulo = tipoFuenteTitulo; } public String getTipoFuenteTitulo() { return
tipoFuenteTitulo; }public void setColorFuenteTitulo(String colorFuenteTitulo) {
this.colorFuenteTitulo = colorFuenteTitulo; } public String getColorFuenteTitulo() { return
colorFuenteTitulo; }public void setTamanioFuenteTitulo(String tamanioFuenteTitulo) {
this.tamanioFuenteTitulo = tamanioFuenteTitulo;} public String getTamanioFuenteTitulo() {return
tamanioFuenteTitulo;}public void setEstiloFuenteTitulo(String estiloFuenteTitulo) {this.estiloFuenteTitulo
= estiloFuenteTitulo; } public String getEstiloFuenteTitulo() { return estiloFuenteTitulo; } public void
setTipoFuenteSubtitulo(String tipoFuenteSubtitulo) { this.tipoFuenteSubtitulo = tipoFuenteSubtitulo;
}public String getTipoFuenteSubtitulo() { return tipoFuenteSubtitulo; }public void
setColorFuenteSubtitulo(String colorFuenteSubtitulo) {this.colorFuenteSubtitulo = colorFuenteSubtitulo;
} public String getColorFuenteSubtitulo() {return colorFuenteSubtitulo; }public void
```

```
setTamanioFuenteSubtitulo(String tamanioFuenteSubtitulo) { this.tamanioFuenteSubtitulo = tamanioFuenteSubtitulo; }
public String getTamanioFuenteSubtitulo() { return tamanioFuenteSubtitulo; } public void setEstiloFuenteSubtitulo(String estiloFuenteSubtitulo) { this.estiloFuenteSubtitulo = estiloFuenteSubtitulo; }public String getEstiloFuenteSubtitulo() { return estiloFuenteSubtitulo; }public void setColorFondoPagina(String colorFondoPagina) {this.colorFondoPagina = colorFondoPagina; } public String getColorFondoPagina() {return colorFondoPagina; }public void setColorFondoFormulario(String colorFondoFormulario) { this.colorFondoFormulario = colorFondoFormulario; }
public String getColorFondoFormulario() {return colorFondoFormulario; } public void setImagenFondo(String imagenFondo) { this.imagenFondo = imagenFondo; } public String getImagenFondo() { return imagenFondo; }public void setTipoFuenteCombo(String tipoFuenteCombo) {this.tipoFuenteCombo = tipoFuenteCombo; } public String getTipoFuenteCombo() { return tipoFuenteCombo; } public void setColorFuenteCombo(String colorFuenteCombo) { this.colorFuenteCombo = colorFuenteCombo; }public String getColorFuenteCombo() { return colorFuenteCombo; }public void setColorFondoCombo(String colorFondoCombo) {this.colorFondoCombo = colorFondoCombo; } public String getColorFondoCombo() {return colorFondoCombo; } public void setTipoFuenteLista(String tipoFuenteLista) { this.tipoFuenteLista = tipoFuenteLista; }public String getTipoFuenteLista() { return tipoFuenteLista; } public void setColorFuenteLista(String colorFuenteLista) {
this.colorFuenteLista = colorFuenteLista; }public String getColorFuenteLista() { return colorFuenteLista; } public void setColorFondoLista(String colorFondoLista) { this.colorFondoLista = colorFondoLista; } public String getColorFondoLista() { return colorFondoLista;} public void setTipoFuenteBoton(String tipoFuenteBoton) { this.tipoFuenteBoton = tipoFuenteBoton; } public String getTipoFuenteBoton() { return tipoFuenteBoton; } public void setColorFuenteBoton(String colorFuenteBoton)
this.colorFuenteBoton = colorFuenteBoton; } public String getColorFuenteBoton() {
return colorFuenteBoton; } public void setColorFondoBoton(String colorFondoBoton) {this.colorFondoBoton = colorFondoBoton; } public String getColorFondoBoton() { return colorFondoBoton; } public void setImagenHeader(String imagenHeader) { this.imagenHeader = imagenHeader; } public String getImagenHeader() { return imagenHeader; } public void setImagenBody(String imagenBody) { this.imagenBody = imagenBody; } public String getImagenBody() {return imagenBody; } public void setImagenFooter(String imagenFooter) { this.imagenFooter = imagenFooter; } public String getImagenFooter() { return imagenFooter; } public void setImagenFoto(int imagenFoto) { this.ImagenFoto = imagenFoto; } public int getImagenFoto() { return ImagenFoto; }}
```

Subtitulo.java

Descripción: Se declara los campos que va a ser usado por el usuario, el codigo del subtitulo, el codigo del tipo de fuente, el codigo del color de la fuente, el codigo del tamaño de la fuente, y el estilo.

Métodos: Get y Set.

Librerías Utilizadas: com.bvg.bean

```
public class Subtitulo { private int codigoSubtitulo;  private String codigoTipoFuente;
private String codigoColorFuente;  private String codigoTamanioFuente;private String estilo; public
Subtitulo() { } public void setCodigoSubtitulo(int codigoSubtitulo) { this.codigoSubtitulo =
codigoSubtitulo; }public int getCodigoSubtitulo() {return codigoSubtitulo;} public void
setCodigoTipoFuente(String codigoTipoFuente) {  this.codigoTipoFuente = codigoTipoFuente;  }
public String getCodigoTipoFuente() {  return codigoTipoFuente; }public void
setCodigoColorFuente(String codigoColorFuente) {this.codigoColorFuente = codigoColorFuente;  }
public String getCodigoColorFuente() {return codigoColorFuente; }public void
setCodigoTamanioFuente(String codigoTamanioFuente) {  this.codigoTamanioFuente =
codigoTamanioFuente;}public String getCodigoTamanioFuente() {  return codigoTamanioFuente; }
public void setEstilo(String estilo) { this.estilo = estilo;} public String getEstilo() {  return estilo;  }}
```

TamanioFuente.java

Descripción: Se declara los campos que va a ser usado por el usuario, el codigo del tamaño de la fuente y valor numerico del tamaño de la fuente.

Métodos: Get y Set.

Librerías Utilizadas: com.bvg.bean

```
public class TamanioFuente {  private String codigoTamanioFuente; private String tamnioFuente; public
TamanioFuente() { }public void setCodigoTamanioFuente(String codigoTamanioFuente) {
this.codigoTamanioFuente = codigoTamanioFuente;  }public String getCodigoTamanioFuente() { return
codigoTamanioFuente; }public void setTamnioFuente(String tamnioFuente) {this.tamnioFuente =
tamnioFuente; }  public String getTamnioFuente() { return tamnioFuente; }
```

TipoFuente.java

Descripción: Se declara los campos que va a ser usado por el usuario, el codigo del tipo de la fuente y valor numerico del tipo de la fuente.

Métodos: Get y Set.

Librerías Utilizadas: com.bvg.bean

```
public class TipoFuente {  private String codigoTipoFuente; private String tipoFuente; public
TipoFuente() {  }public void setCodigoTipoFuente(String codigoTipoFuente) { this.codigoTipoFuente =
codigoTipoFuente; } public String getCodigoTipoFuente() {  return codigoTipoFuente; }public void
setTipoFuente(String tipoFuente) { this.tipoFuente = tipoFuente;  } public String getTipoFuente() {
return tipoFuente;  }}
```

Titulo.java

Descripción: Se declara los campos que va a ser usado por el usuario, el codigo del titulo, el codigo del color de la fuente, el codigo del tipo de la fuente, el codigo del tamaño de la fuente y su estilo.

Métodos: Get y Set.

Librerías Utilizadas: com.bvg.bean

```
public class Titulo {    private int codigoTitulo;    private String codigoColorFuente;    private String
codigoTipoFuente;    private String codigoTamanioFuente;    private String estilo;    public Titulo() {    }
public void setCodigoColorFuente(String codigoColorFuente) {        this.codigoColorFuente =
codigoColorFuente;  }   public String getCodigoColorFuente() {return codigoColorFuente;  }public void
setCodigoTamanioFuente(String codigoTamanioFuente) {              this.codigoTamanioFuente =
codigoTamanioFuente;   }
public String getCodigoTamanioFuente() {   return codigoTamanioFuente; }
public void setCodigoTitulo(int codigoTitulo) {     this.codigoTitulo = codigoTitulo;
}public int getCodigoTitulo() {return codigoTitulo;}public void setCodigoTipoFuente(String
codigoTipoFuente) {this.codigoTipoFuente = codigoTipoFuente; } public String getCodigoTipoFuente()
{return codigoTipoFuente;     } public void setEstilo(String estilo) { this.estilo = estilo;} public String
getEstilo() {     return estilo; }}
```

🗁**BO**

Realizan la conexión a la base de datos.

ColorFuenteBO.java

Descripción: Se declara la conexión a la base de datos, mediante un vector que busca el color de la fuente.

Enlaces: ColorFuenteDAO.

Librerías Utilizadas: com.bvg.dao

com.bvg.util

```
import com.bvg.dao.ColorFuenteDAO;
import com.bvg.util.Base;
import java.sql.Connection;
import java.util.Vector;
public class ColorFuenteBO {
    private Connection con = null;
    public ColorFuenteBO() {    }    public Vector buscarTodos() {
        Vector valorDevolver = new Vector();        con = Base.getCon();
        ColorFuenteDAO unColorFuenteDAO = new ColorFuenteDAO(con);
        try {      valorDevolver = unColorFuenteDAO.buscarTodos();
            con.commit();     } catch (Throwable e) {    System.out.println(e);   try {
                con.rollback();       } catch (Throwable ee) {   System.out.println(ee);      }     }
        return valorDevolver;    }}
```

EtiquetaBO.java

Descripción: Se declara la conexión a la base de datos, mediante dos metodos, uno busca el codigo de la etiqueta BuscarCodigo, y el otro la guarda Guardar.

Enlaces: EtiquetaDAO, PersonaDAO.

Librerías Utilizadas: com.bvg.bean
com.bvg.bean
com.bvg.dao
com.bvg.util

```
import com.bvg.bean.Etiqueta;
import com.bvg.dao.EtiquetaDAO;
import com.bvg.dao.PersonaDAO;
import com.bvg.util.Base;
import java.sql.Connection;
public class EtiquetaBO {    private Connection con = null;
    public EtiquetaBO() {    }
    public int buscarCodigo() {    int codigo = 0;
        con = Base.getCon();   EtiquetaDAO unaEtiquetaDAO = new EtiquetaDAO(con);
        try {  codigo = unaEtiquetaDAO.buscarCodigo();  con.commit();
        } catch (Throwable e) {  System.out.println(e);   try {    con.rollback();} catch (Throwable ee) {
                System.out.println(ee);      }       }    return codigo;   }
    public boolean guardar(Etiqueta unaNuevaEtiqueta) {
        boolean valorDevolver = false;    con = Base.getCon();
        EtiquetaDAO unaEtiquetaDAO = new EtiquetaDAO(con);
        try { valorDevolver = unaEtiquetaDAO.guardar(unaNuevaEtiqueta);  con.commit();
        } catch (Throwable e) { System.out.println(e);  try {  con.rollback();
            } catch (Throwable ee) {  System.out.println(ee);      }    }   return valorDevolver;   }
```

```
public int compararDatosEtiqueta(String tipoFuenteEtiqueta,  String colorFuenteEtiqueta,
String tamanioFuenteEtiqueta) {   int codigo = 0;    con = Base.getCon();
   EtiquetaDAO unaEtiquetaDAO = new EtiquetaDAO(con);     try {
      codigo = unaEtiquetaDAO.compararDatosEtiqueta(tipoFuenteEtiqueta, colorFuenteEtiqueta,
                                    tamanioFuenteEtiqueta);
      con.commit();  } catch (Throwable e) {     System.out.println(e);      try {
         con.rollback();   } catch (Throwable ee) {      System.out.println(ee);       }      }
   return codigo;    }}
```

FondoPaginaBO.java

Descripción: Se declara la conexión a la base de datos, mediante dos metodos, uno busca el codigo del fondo de la pagina BuscarCodigo, y el otro la guarda Guardar.

Enlaces: FondoPaginaDAO.

Bean: FondoPagina

```
package com.bvg.bo;
import com.bvg.bean.FondoPagina;
import com.bvg.dao.FondoPaginaDAO;
import com.bvg.dao.TituloDAO;
import com.bvg.util.Base;
import java.sql.Connection;
public class FondoPaginaBO {   private Connection con = null;   public FondoPaginaBO() {
   }     public int buscarCodigo() {       int codigo = 0;     con = Base.getCon();       FondoPaginaDAO
unFondoPaginaDAO = new FondoPaginaDAO(con);      try {
      codigo = unFondoPaginaDAO.buscarCodigo(); con.commit();     } catch (Throwable e) {
      System.out.println(e);   try {   con.rollback(); } catch (Throwable ee) {
         System.out.println(ee);      }     }     return codigo;   }
public boolean guardar(FondoPagina unFondoPagina) {
boolean valorDevolver = false;    con = Base.getCon();
FondoPaginaDAO unaEtiquetaDAO = new FondoPaginaDAO(con);     try {
valorDevolver = unaEtiquetaDAO.guardar(unFondoPagina);    con.commit();
} catch (Throwable e) {    System.out.println(e);       try {      con.rollback();
 } catch (Throwable ee) {      System.out.println(ee);       }       }
   return valorDevolver;    }
  public int compararDatosFondoPagina(String unCodigoColorFondo) {
   int codigo = 0;       con = Base.getCon();
   FondoPaginaDAO unaTituloDAO = new FondoPaginaDAO(con);    try {
      codigo = unaTituloDAO.compararDatosFondoPagina(unCodigoColorFondo);
      con.commit();     } catch (Throwable e) {    System.out.println(e);       try {
         con.rollback();     } catch (Throwable ee) { System.out.println(ee);      }  }
   return codigo;  }}
```

PersonaBO.java

Descripción: Se declara la conexión a la base de datos, mediante tres metodos, uno busca el codigo de la plantilla BuscarCodigoPlantilla, el segundo valida el acceso al usuario mediante el login y la contraseña ValidaUsuario y el otro la guarda la Plantilla GuardarPlantilla.

Enlaces: PersonaDAO.

Bean: Persona

```
package com.bvg.bo;
import com.bvg.bean.Persona;
import com.bvg.dao.PersonaDAO;
import com.bvg.util.Base;
import java.sql.Connection;
public class PersonaBO {
private Connection con = null;
public boolean guardarPlantilla(String usuario, int codigoPlantilla) {
boolean valorDevolver = false;
con = Base.getCon();
PersonaDAO unaPersonaDAO = new PersonaDAO(con);
try { valorDevolver =     unaPersonaDAO.guardarPlantilla(usuario, codigoPlantilla);
con.commit(); } catch (Throwable e) { System.out.println(e);   try {       con.rollback();
} catch (Throwable ee) {      System.out.println(ee);     }     }     return valorDevolver;   }
public boolean validaUsuario(String unUsuario, String unaClave) {
 boolean  valorDevolver  =  false;           con  =  Base.getCon();PersonaDAO  unaPersonaDAO  =  new
PersonaDAO(con);    try {
valorDevolver = unaPersonaDAO.validaUsuario(unUsuario, unaClave);      con.commit();
       } catch (Throwable e) {   System.out.println(e);        try {       con.rollback();
           } catch (Throwable ee) {        System.out.println(ee);      }    }      return valorDevolver;   }
public String validaUsuario2(String unUsuario, String unaClave) {     String Plantilla = null;
con = Base.getCon();  PersonaDAO unaPersonaDAO = new PersonaDAO(con);     try {
Plantilla = unaPersonaDAO.validaUsuario2(unUsuario, unaClave);      con.commit();
} catch (Throwable e) {    System.out.println(e);      try {       con.rollback();   } catch (Throwable ee) {
              System.out.println(ee);       }   }   return Plantilla;  }
    public int buscarCodigoPlantilla(String unUsuario, String unaClave) {
       int   codigoPlantilla   =   0;        con   =   Base.getCon();   PersonaDAO   unaPersonaDAO   =   new
PersonaDAO(con); try { codigoPlantilla =
unaPersonaDAO.buscarCodigoPlantilla(unUsuario, unaClave);   con.commit();          } catch (Throwable
e) {    System.out.println(e);     try {   con.rollback();     } catch (Throwable ee) {
              System.out.println(ee);       }    }   return codigoPlantilla;  }
    public int buscarCodigoPlantillaUser(String unUsuario) {
       int   codigoPlantilla   =   0;            con   =   Base.getCon();   PersonaDAO   unaPersonaDAO   =   new
PersonaDAO(con);  try {codigoPlantilla =
```

```
            unaPersonaDAO.buscarCodigoPlantillaUser(unUsuario);
        con.commit();     } catch (Throwable e) {  System.out.println(e);     try {      con.rollback();
        } catch (Throwable ee) {     System.out.println(ee);    }    }   return codigoPlantilla;   }
public Persona buscarUsuario(String unUsuario) {      Persona unaPersona = new Persona();
      con = Base.getCon();     PersonaDAO unaPersonaDAO = new PersonaDAO(con);     try {
unaPersona = unaPersonaDAO.buscarUsuario(unUsuario);     con.commit();  } catch (Throwable e) {
        System.out.println(e);              try    {       con.rollback();    catch    (Throwable    ee)    {
System.out.println(ee);         }      }      return unaPersona;   }
   public boolean guardar2(Persona unaPersona) {      boolean valorDevolver = false;
      con = Base.getCon();    PersonaDAO unaPersonaDAO = new PersonaDAO(con);     try {
        valorDevolver = unaPersonaDAO.guardar2(unaPersona);         if (valorDevolver == false) {
           System.out.println("nooooooooooooooooo: ");         }
        con.commit();     } catch (Throwable e) {   System.out.println(e);     try {          con.rollback();
        } catch (Throwable ee) {       System.out.println(ee);      }      }     return valorDevolver;   }}
```

PlantillaBO.java

Descripción: Se declara la conexión a la base de datos, mediante tres metodos, uno busca el codigo de la plantilla BuscarPlantilla, el segundo busca el estilo de la plantilla BuscarEstiloPlantilla y el otro la guarda la Plantilla Nueva en Guardar.

Enlaces: PersonaDAO, PlantillaDAO.

Bean: Persona, Plantilla, PlantillaCodigo.

```
package com.bvg.bo;
import com.bvg.dao.PlantillaDAO;
import com.bvg.util.Base;
import com.bvg.bean.PlantillaCodigo;
import com.bvg.bean.Plantilla;
import com.bvg.bean.Persona;
import com.bvg.dao.ColorFondoDAO;
import com.bvg.dao.PersonaDAO;
import java.sql.Connection;
import java.util.Vector;
public class PlantillaBO {
   private Connection con = null;    public PlantillaBO() {    }
   public Plantilla buscarPlantilla(int codigoPlantilla) {      Plantilla unaPlantilla = new Plantilla();
      con = Base.getCon();    PlantillaDAO unaPlantillaDAO = new PlantillaDAO(con);      try {
         unaPlantilla = unaPlantillaDAO.buscarPlantilla(codigoPlantilla);      con.commit();
      } catch (Throwable e) {     System.out.println(e);  try {  con.rollback();
         } catch (Throwable ee) {    System.out.println(ee);    }    }   return unaPlantilla; }
```

```
public PlantillaCodigo recuperarPlantilla(int unCodigoPlantilla) {
PlantillaCodigo unaPlantillaCodigo = new PlantillaCodigo();    con = Base.getCon();
        PlantillaDAO unaPlantillaDAO = new PlantillaDAO(con);   try {     unaPlantillaCodigo =
                    unaPlantillaDAO.recuperarPlantilla(unCodigoPlantilla);        con.commit();
        } catch (Throwable e) {  System.out.println(e);  try {    con.rollback();
            } catch (Throwable ee) {  System.out.println(ee);     }    }     return unaPlantillaCodigo;   }
    public int buscarCodigo() {     int codigo = 0;    con = Base.getCon();
        PlantillaDAO unaPlantillaDAO = new PlantillaDAO(con);     try {
            codigo = unaPlantillaDAO.buscarCodigo(); con.commit();    } catch (Throwable e) {
            System.out.println(e);         try {     con.rollback();       } catch (Throwable ee) {
                System.out.println(ee);       }     }     return codigo;  }
public int buscarEstiloPlantilla(String unUsuario) {    int valorRetornar = 0;
        con = Base.getCon();   PlantillaDAO unaPlantillaDAO = new PlantillaDAO(con);     try {
            valorRetornar = unaPlantillaDAO.buscarEstiloPlantilla(unUsuario);       con.commit();
        } catch (Throwable e) {  System.out.println(e);    try {   con.rollback();
            } catch (Throwable ee) {     System.out.println(ee);     }  } return valorRetornar;    }
public boolean guardar(PlantillaCodigo unaNuevaPlantilla) {     boolean valorDevolver = false;
        con = Base.getCon();PlantillaDAO unaPlantillaDAO = new PlantillaDAO(con);     try {
            valorDevolver = unaPlantillaDAO.guardar(unaNuevaPlantilla);       con.commit();
        } catch (Throwable e) {System.out.println(e);     try {     con.rollback();    } catch (Throwable ee) {
                System.out.println(ee);      }   }  return valorDevolver;}
    public int buscarCodigoPlantilla(String unUsuario) {     int codigo = 0;    con = Base.getCon();
        PlantillaDAO unaPlantillaDAO = new PlantillaDAO(con);      try {
            codigo = unaPlantillaDAO.buscarCodigoPlantilla(unUsuario);
            con.commit();     } catch (Throwable e) {    System.out.println(e);          try {
                con.rollback();   } catch (Throwable ee) {       System.out.println(ee);    }     }
        return codigo; }    public boolean actualizar(PlantillaCodigo unaNuevaPlantilla) {
        boolean valorDevolver = false;       con = Base.getCon();       PlantillaDAO unaPlantillaDAO = new
PlantillaDAO(con);    try {       valorDevolver = unaPlantillaDAO.actualizar(unaNuevaPlantilla);
            con.commit();    } catch (Throwable e) {    System.out.println(e);    try {       con.rollback();
            } catch (Throwable ee) {   System.out.println(ee);   }  } return valorDevolver;    }
public Vector buscarTodos() { Vector valorDevolver = new Vector();    con = Base.getCon();
        PlantillaDAO    unaPlantillaDAO    =    new    PlantillaDAO(con);          try    {    valorDevolver    =
unaPlantillaDAO.buscarTodos();       con.commit();  catch  (Throwable  e)  {        System.out.println(e);
try  {      con.rollback();     }  catch  (Throwable  ee)  {       System.out.println(ee);         }        }      return
valorDevolver;  }
    public Plantilla recuperaCodigo(int codigoPlantilla) {     Plantilla unaPlantilla = new Plantilla();
        con = Base.getCon();  PlantillaDAO unaPlantillaDAO = new PlantillaDAO(con);
        try {       unaPlantilla = unaPlantillaDAO.recuperaCodigo(codigoPlantilla);       con.commit();
        } catch (Throwable e) {     System.out.println(e);    try {     con.rollback();    } catch (Throwable ee) {
                System.out.println(ee);       }    } return unaPlantilla; }}
```

SubtituloBO.java

Descripción: Se declara la conexión a la base de datos, mediante dos metodos, uno busca el codigo del subtitulo BuscarCodigo, y el otro guarda el Subtitulo en Guardar.

Enlaces: SubtituloDAO.

Bean: Subtitulo.

```
package com.bvg.bo;
import com.bvg.bean.Subtitulo;
import com.bvg.dao.SubtituloDAO;
import com.bvg.dao.TituloDAO;
import com.bvg.util.Base;
import java.sql.Connection;
public class SubtituloBO {
private Connection con = null;
public SubtituloBO() {}  public int buscarCodigo() {    int codigo = 0;       con = Base.getCon();
       SubtituloDAO unSubtituloDAO = new SubtituloDAO(con);    try {
           codigo = unSubtituloDAO.buscarCodigo();  con.commit();
       } catch (Throwable e) { System.out.println(e);        try {       con.rollback();
           } catch (Throwable ee) {     System.out.println(ee);     } } return codigo;    }
   public boolean guardar(Subtitulo unNuevoSubtitulo) {     boolean valorDevolver = false;
       con = Base.getCon();     SubtituloDAO unSubtituloDAO = new SubtituloDAO(con);
       try {        valorDevolver = unSubtituloDAO.guardar(unNuevoSubtitulo);        con.commit();
       } catch (Throwable e) {      System.out.println(e);      try {     con.rollback();          } catch (Throwable
ee) {          System.out.println(ee);      }      }    return valorDevolver;   }
   public    int    compararDatosTitulo(String    tipoFuenteSubtitulo,            String    colorFuenteSubtitulo,
String  tamanioFuenteSubtitulo)  {        int  codigo  =  0;        con  =  Base.getCon();        SubtituloDAO
unaTituloDAO     =     new     SubtituloDAO(con);                      try     {                         codigo     =
unaTituloDAO.compararDatosSubtitulo(tipoFuenteSubtitulo,                          colorFuenteSubtitulo,
tamanioFuenteSubtitulo);       con.commit();     } catch  (Throwable  e) {    System.out.println(e);       try {
con.rollback();  } catch (Throwable ee) {      System.out.println(ee);     } }  return codigo;}}
```

TamanioFuenteBO.java

Descripción: Se declara la conexión a la base de datos, mediante un vector que busca el tamaño de la fuente.

Enlaces: TamanioFuenteDAO.

```
package com.bvg.bo;
import com.bvg.bean.Subtitulo;
import com.bvg.dao.SubtituloDAO;
import com.bvg.dao.TituloDAO;
import com.bvg.util.Base;
import java.sql.Connection;
public class SubtituloBO {  private Connection con = null;    public SubtituloBO() {   }
public int buscarCodigo() {     int codigo = 0;     con = Base.getCon();      SubtituloDAO unSubtituloDAO
= new SubtituloDAO(con);     try { codigo = unSubtituloDAO.buscarCodigo();      con.commit();
       } catch (Throwable e) {       System.out.println(e);    try {   con.rollback();       } catch (Throwable ee)
{     System.out.println(ee);   } }   return codigo;   }
```

```
public boolean guardar(Subtitulo unNuevoSubtitulo) {      boolean valorDevolver = false;         con = Base.getCon();        SubtituloDAO unSubtituloDAO = new SubtituloDAO(con);        try {
        valorDevolver = unSubtituloDAO.guardar(unNuevoSubtitulo);             con.commit();        } catch (Throwable e) {              System.out.println(e);              try {                con.rollback();            } catch (Throwable ee) {   System.out.println(ee);      }    }  return valorDevolver;  }
public int compararDatosTitulo(String tipoFuenteSubtitulo, String colorFuenteSubtitulo, String tamanioFuenteSubtitulo) {        int codigo = 0;       con = Base.getCon();          SubtituloDAO unaTituloDAO = new SubtituloDAO(con);          try { codigo = unaTituloDAO.compararDatosSubtitulo(tipoFuenteSubtitulo, colorFuenteSubtitulo, tamanioFuenteSubtitulo); con.commit();} catch (Throwable e) {   System.out.println(e);     try {
          con.rollback();      } catch (Throwable ee) {      System.out.println(ee);      }       }        return codigo;    }}
```

TipoFuenteBO.java

Descripción: Se declara la conexión a la base de datos, mediante un vector que busca el tipo de la fuente BuscarTodos.

Enlaces: TipoFuenteDAO.

```
package com.bvg.bo;
import com.bvg.dao.TipoFuenteDAO;
import com.bvg.util.Base;
import java.sql.Connection;
import java.util.Vector;
public class TipoFuenteBO {
private Connection con = null;
public TipoFuenteBO() {  }   public Vector buscarTodos() {     Vector valorDevolver = new Vector();
       con = Base.getCon();    TipoFuenteDAO unColorFuenteDAO = new TipoFuenteDAO(con);
       try {        valorDevolver = unColorFuenteDAO.buscarTodos();        con.commit();
       } catch (Throwable e) {        System.out.println(e);        try {             con.rollback();
          } catch (Throwable ee) {        System.out.println(ee);       }     }       return valorDevolver;
   }}
```

TituloBO.java

Descripción: Se declara la conexión a la base de datos, mediante dos metodos, uno busca el codigo del titulo BuscarCodigo, y el otro guarda el titulo en Guardar.

Enlaces: TituloDAO.

Bean: Titulo.

```
package com.bvg.bo;
import com.bvg.bean.Titulo;
import com.bvg.dao.EtiquetaDAO;
import com.bvg.dao.TituloDAO;
import com.bvg.util.Base;
import java.sql.Connection;
public class TituloBO {
private Connection con = null; public TituloBO() { }   public int buscarCodigo() {      int codigo = 0;
      con = Base.getCon();    TituloDAO unTituloDAO = new TituloDAO(con);     try {
         codigo = unTituloDAO.buscarCodigo();      con.commit();    } catch (Throwable e) {
         System.out.println(e);            try   {            con.rollback();  }  catch  (Throwable  ee)  {
System.out.println(ee);   }     } return codigo;   }   public boolean guardar(Titulo unNuevoTitulo) {
      boolean valorDevolver = false;      con = Base.getCon();       TituloDAO unTituloDAO = new
TituloDAO(con);    try {      valorDevolver = unTituloDAO.guardar(unNuevoTitulo);        con.commit();
      } catch (Throwable e) {  System.out.println(e);   try { con.rollback();   } catch (Throwable ee) {
            System.out.println(ee);  }   }  return valorDevolver;  }
   public   int   compararDatosTitulo(String   tipoFuenteTitulo,              String   colorFuenteTitulo,
String tamanioFuenteTitulo) {      int codigo = 0;   con = Base.getCon();   TituloDAO unaTituloDAO =
new  TituloDAO(con);       try  {       codigo  =  unaTituloDAO.compararDatosTitulo(tipoFuenteTitulo,
colorFuenteTitulo,         tamanioFuenteTitulo);con.commit();        }     catch     (Throwable     e)    {
System.out.println(e);              try  {            con.rollback();            }  catch  (Throwable  ee)  {
System.out.println(ee);        }    }      return codigo;   }}
```

🗁DAO

ColorFondoDAO.java

Descripción: Se declara las sentencias SQL a la base de datos, mediante un vector que busca el color de fondo, BuscarTodos.

Bean: ColorFondo, ColorFuente.

Donde se utilizan: EstiloPropio.jsp

Librerías:

java.sql.Connection;import java.sql.PreparedStatement;
java.sql.ResultSet;import java.sql.SQLException;import java.util.Vector;
java.lang.String;com.bvg.bean.ColorFondo;com.bvg.bean.Colorfuente;

```
Recibe la conexión de la Base de Datos
private Connection con = null;   public ColorFondoDAO(Connection unaCon) {
con = unaCon;  }

Busca todos los colores de fondo guardados en la base de datos
    public Vector buscarTodos() throws Throwable {
        Vector valorDevolver = new Vector();
        String sql = "SELECT * FROM colorfondo ORDER BY descripcion";
        PreparedStatement unPs = con.prepareStatement(sql);
        ResultSet unRs = unPs.executeQuery();
        ColorFondo unColorFondo;
        while (unRs.next()) {
            unColorFondo = new ColorFondo();
            unColorFondo.setCodigoColorFondo(unRs.getString("codigocolor"));
            unColorFondo.setDescripcion(unRs.getString("descripcion"));
            valorDevolver.addElement(unColorFondo);        }        unRs.close();     unPs.close();         return
valorDevolver;    }}
```

ColorFuenteDAO.java

Descripción: Se declara las sentencias SQL a la base de datos, mediante un vector que busca el color de la fuente, BuscarTodos.

Bean: ColorFuente.

Donde se utilizan: EstiloPropio.jsp

```
package com.bvg.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Vector;
import java.lang.String;
import com.bvg.bean.Colorfuente;
public class ColorFuenteDAO {
Recibe la conexión de la Base de Datos
   private Connection con = null;   public ColorFuenteDAO(Connection unaCon) {        con = unaCon;    }
Busca todos los colores de fuentes dados en la base de datos

    public Vector buscarTodos() throws Throwable {        Vector valorDevolver = new Vector();
        String sql = "SELECT * FROM colorfuente ORDER BY nombrecolor";           PreparedStatement
unPs = con.prepareStatement(sql);        ResultSet unRs = unPs.executeQuery();
        Colorfuente unColorFuente;      while (unRs.next()) {         unColorFuente = new Colorfuente();
            unColorFuente.setCodigocolorFuente(unRs.getString("codigocolorfuente"));
            unColorFuente.setNombreColor(unRs.getString("nombrecolor"));
unColorFuente.setRgb(unRs.getString("rgb")); valorDevolver.addElement(unColorFuente);     }
        unRs.close();     unPs.close();     return valorDevolver;  }}
```

EtiquetaDAO.java

Descripción: Se declara las sentencias SQL a la base de datos, mediante tres metodos, uno busca el codigo de la etiqueta BuscarCodigo, el otro BuscaTodos y el otro la guarda la etiqueta en Guardar.

Bean: Etiqueta, Persona.

Donde se utilizan: EstiloPropio.jsp

```
package com.bvg.dao;
import com.bvg.bean.Etiqueta;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Vector;
public class EtiquetaDAO {

Recibe la Conexión de la base de Datos
    private Connection con = null;
    public EtiquetaDAO(Connection unaCon) {
        con = unaCon;
    }
Busca todos los colores de fuentes dados en la base de datos

    public Vector buscarTodos() throws Throwable {
        Vector valorDevolver = new Vector();
        String sql = "SELECT * FROM etiqueta";
        PreparedStatement unPs = con.prepareStatement(sql);
        ResultSet unRs = unPs.executeQuery();
        Etiqueta unaEtiqueta;
        while (unRs.next()) {
            unaEtiqueta = new Etiqueta();
            unaEtiqueta.setCodigoEtiqueta(unRs.getInt("codigoetiqueta"));
            unaEtiqueta.setCodigoTipoFuente(unRs.getString("codigotipofuente"));
            unaEtiqueta.setCodigoColorFuente(unRs.getString("codigocolorfuente"));
            valorDevolver.addElement(unaEtiqueta);        }        unRs.close();        unPs.close();
        return valorDevolver;    }    public int buscarCodigo() throws Throwable {        int codigo = 0;
        String sql = "SELECT max(codigoetiqueta)as codigoetiqueta FROM etiqueta";
        PreparedStatement unPs = con.prepareStatement(sql);ResultSet unRs = unPs.executeQuery();
        while (unRs.next()) {    codigo = unRs.getInt("codigoetiqueta");     }unRs.close();    unPs.close();
 return codigo;  }  public boolean guardar(Etiqueta unaEtiqueta) throws Throwable {       String sql =
            "INSERT        INTO        etiqueta        (codigoetiqueta,codigotipofuente,        codigocolorfuente,
codigotamaniofuente) values (?,?,?,?)";    PreparedStatement unPs = con.prepareStatement(sql);
unPs.setInt(1, unaEtiqueta.getCodigoEtiqueta());
        unPs.setString(2, unaEtiqueta.getCodigoTipoFuente());
        unPs.setString(3, unaEtiqueta.getCodigoColorFuente());
        unPs.setString(4, unaEtiqueta.getCodigoTamanioFuente());
        int i = unPs.executeUpdate();     unPs.close();      if (i == 1)   return true;     else      return false;  }
    public int compararDatosEtiqueta(String tipoFuenteEtiqueta,
```

```
String colorFuenteEtiqueta, String tamanioFuenteEtiqueta) throws Throwable {       int codigo = 0;
       String sql =      "select codigoetiqueta from etiqueta where codigotipofuente = ? And" +
          "  codigocolorfuente  =  ?  and  codigotamaniofuente  =  ?";PreparedStatement  unPs  =
con.prepareStatement(sql);                    unPs.setString(1,       tipoFuenteEtiqueta);unPs.setString(2,
colorFuenteEtiqueta);               unPs.setString(3,  tamanioFuenteEtiqueta);          ResultSet  unRs  =
unPs.executeQuery();          while   (unRs.next())  {            if  (unRs.isFirst())  {              codigo  =
unRs.getInt("codigoetiqueta");          }    }    unRs.close();   unPs.close();    return codigo;   }}
```

FondoPaginaDAO.java

Descripción: Se declara las sentencias SQL a la base de datos, mediante tres metodos, uno busca el codigo del fondo de la pagina BuscarCodigo, y el otro busca todos los fondos de pagina y el otro guarda el fondo de la pagina en Guardar.

Bean: FondoPagina

Donde se utilizan: EstiloPropio.jsp

```
package com.bvg.dao;
import com.bvg.bean.FondoPagina;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Vector;
public class FondoPaginaDAO {

Recibe la conexión de la Base de Datos
   private Connection con = null;
   public FondoPaginaDAO(Connection unaCon) {       con = unaCon;    }

Busca todos los fondos de páginas dados en la base de datos

   public Vector buscarTodos() throws Throwable {       Vector valorDevolver = new Vector();
      String   sql   =   "SELECT   *   FROM   fondopagina";                     PreparedStatement   unPs   =
con.prepareStatement(sql);    ResultSet unRs = unPs.executeQuery();  FondoPagina unFondoPagina;
      while (unRs.next()) {       unFondoPagina = new FondoPagina();
         unFondoPagina.setCodigoFondoPagina(unRs.getInt("codigofondopagina"));
         unFondoPagina.setCodigoColorFondo(unRs.getString("codigocolor"));
valorDevolver.addElement(unFondoPagina);     }  unRs.close();  unPs.close(); return valorDevolver;   }
public int buscarCodigo() throws Throwable {     int codigo = 0;String sql = "SELECT codigofondopagina
FROM  fondopagina";    PreparedStatement  unPs  =  con.prepareStatement(sql);ResultSet  unRs  =
unPs.executeQuery();              while     (unRs.next())     {         if     (unRs.isLast())     {codigo     =
unRs.getInt("codigofondopagina");   }    } unRs.close();   unPs.close();    return codigo;  }
public boolean guardar(FondoPagina unFondoPagina) throws Throwable {
      String sql =        "INSERT INTO fondopagina (codigofondopagina,codigocolor) values (?,?)";
```

```
        PreparedStatement unPs = con.prepareStatement(sql);
        unPs.setInt(1, unFondoPagina.getCodigoFondoPagina());
        unPs.setString(2, unFondoPagina.getCodigoColorFondo());
        int i = unPs.executeUpdate();     unPs.close();    if (i == 1)  return true;    else     return false;   }
public int compararDatosFondoPagina(String unCodigoColorFondo) throws Throwable {
int codigo = 0; String sql =       "select codigofondopagina from fondopagina where codigocolor = ?";
PreparedStatement unPs = con.prepareStatement(sql);    unPs.setString(1, unCodigoColorFondo);
ResultSet unRs = unPs.executeQuery(); while (unRs.next()) {  if (unRs.isFirst()) {
codigo = unRs.getInt("codigofondopagina");      }    }     unRs.close();    unPs.close();       return codigo;
    }}
```

PersonaDAO.java

Descripción: Se declara las sentencias SQL a la base de datos, mediante tres metodos, uno busca el codigo de la plantilla BuscarCodigoPlantilla, el segundo valida el acceso al usuario mediante el login y la contraseña ValidaUsuario y el otro la guarda la Plantilla GuardarPlantilla.

Bean: Persona

Donde se utilizan: EstiloPropio.jsp

```
package com.bvg.dao;
import com.bvg.bean.Persona;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
public class PersonaDAO {
private Connection con = null;

Recibe la conexión de la Base de Datos

    public PersonaDAO(Connection unaCon) {      con = unaCon;    }
Metodo para guadar el codigo de una plantilla personalizada creada por el usuario

    public boolean guardarPlantilla(String usuario,        int codigoPlantilla) throws Throwable {
        String sql = "UPDATE usuarios SET codigoplantilla=? WHERE usuario = ?";
        PreparedStatement unPs = con.prepareStatement(sql);        unPs.setInt(1, codigoPlantilla);
        unPs.setString(2, usuario);        int i = unPs.executeUpdate();        unPs.close();
        if (i == 1)           return true;        else           return false;    }
    public boolean validaUsuario(String unUsuario, String unaClave) throws Throwable {
        boolean valorRetornar = false;   String sql =
            "SELECT usuario,claveactual FROM usuarios where usuario = ? AND claveactual = ?";
        PreparedStatement unPs = con.prepareStatement(sql);System.out.println("usuario: " + unUsuario);
        System.out.println("clave   :   "   +   unaClave);System.out.println("Llega   a   valida   Usuario   de
PersonaDAO");    unPs.setString(1, unUsuario);     unPs.setString(2, unaClave);
        ResultSet unRs = unPs.executeQuery(); Creamos result set       if (unRs.next()) {
```

```
        valorRetornar = true;
      } unRs.close(); /* Cerramos result set xxxxxx*/unPs.close(); /* Cerramos Preapared statement */
       return valorRetornar;    }
    public String validaUsuario2(String unUsuario,  String unaClave) throws Throwable {
       String valorRetornar = null;     String sql =
          "SELECT  usuario,claveactual,plantillacss,  codigoplantilla  FROM  usuarios  where  usuario  =  ?
AND claveactual = ?";     PreparedStatement unPs = con.prepareStatement(sql);
       System.out.println("usuario: " + unUsuario);       System.out.println("clave : " + unaClave);
       System.out.println("Llega a valida Usuario de PersonaDAO");      unPs.setString(1, unUsuario);
       unPs.setString(2, unaClave);        ResultSet unRs = unPs.executeQuery(); /*Creamos result set */
       if (unRs.next()) {       valorRetornar = unRs.getString("plantillacss");
          System.out.println("archivo : " + valorRetornar);       }       unRs.close();     unPs.close();
       return valorRetornar;    }
    public int buscarCodigoPlantilla(String unUsuario,  String unaClave) throws Throwable {
       int valorRetornar = 0;     String sql =
          "SELECT  codigoplantilla FROM usuarios where usuario = ? AND claveactual = ?";
       PreparedStatement   unPs   =   con.prepareStatement(sql);   System.out.println("usuario:   "   +
unUsuario);   System.out.println("clave : " + unaClave);         System.out.println("Llega a valida Usuario
de PersonaDAO");
       unPs.setString(1, unUsuario);    unPs.setString(2, unaClave);
       ResultSet unRs = unPs.executeQuery(); /*Creamos result set */
       if              (unRs.next())              {valorRetornar              =              unRs.getInt("codigoplantilla");
System.out.println("codigoPlantilla : " + valorRetornar);       }       unRs.close(); /* Cerramos result set
xxxxxx*/       unPs.close(); /* Cerramos Preapared statement */    return valorRetornar;    }
    public int buscarCodigoPlantillaUser(String unUsuario) throws Throwable {       int valorRetornar = 0;
       String sql = "SELECT  codigoplantilla FROM usuarios where usuario = ?";
       PreparedStatement unPs = con.prepareStatement(sql);System.out.println("usuario: " + unUsuario);
       System.out.println("Llega a valida Usuario de PersonaDAO");     unPs.setString(1, unUsuario);
 ResultSet   unRs   =   unPs.executeQuery();   /*Creamos   result   set   */             if   (unRs.next())   {
valorRetornar = unRs.getInt("codigoplantilla");     System.out.println("plantilla : " + valorRetornar);     }
       unRs.close(); /* Cerramos result set xxxxxx*/
       unPs.close(); /* Cerramos Preapared statement */
 return valorRetornar;} public Persona buscarUsuario(String unUsuario) throws Throwable {
       Persona unaPersona = new Persona();  String sql = "SELECT * FROM usuarios where usuario =
?";       PreparedStatement  unPs  =  con.prepareStatement(sql);       System.out.println("usuario:  "  +
unUsuario);         System.out.println("Llega  a  valida  Usuario  de  PersonaDAO");    unPs.setString(1,
unUsuario);   ResultSet unRs = unPs.executeQuery(); /*Creamos result set */
      if (unRs.next()) {
          unaPersona.setArchivo(unRs.getString("plantillacss"));
unaPersona.setCodigoPlantilla(unRs.getInt("codigoplantilla"));
          unaPersona.setUsuario(unRs.getString("usuario"));
unaPersona.setClave(unRs.getString("claveactual"));       }
      unRs.close(); /* Cerramos result set xxxxxx*/
       unPs.close(); /* Cerramos Preapared statement */
      return unaPersona;   }
    public boolean guardar2(Persona unaPersona) throws Throwable {
 String sql = "update usuarios set plantillacss=? where usuario=?";          PreparedStatement unPs =
con.prepareStatement(sql);
       unPs.setString(2, unaPersona.getArchivo());
  unPs.setString(1, unaPersona.getUsuario());
       int i = unPs.executeUpdate();
  unPs.close();
       if (i == 1) {
      return true;       } else {
       return false;       }    }}
```

PlantillaDAO.java

Descripción: Se declara las sentencias SQL a la base de datos, mediante tres metodos, uno busca el codigo de la plantilla BuscarPlantilla, el segundo recupera el estilo de la plantilla BuscarEstiloPlantilla y el otro la guarda la Plantilla Nueva en Guardar.

Bean: Persona, Plantilla, PlantillaCodigo.

```
package com.bvg.dao;
import com.bvg.bean.Etiqueta;
import com.bvg.bean.Plantilla;
import com.bvg.bean.PlantillaCodigo;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Vector;

CLASE PARA RECUPERAR LA PLANTILLA GUARDADA Y QUE SE COLOQUE EN LA VENTANA
DEL USUARIO

public class PlantillaDAO {
    private Connection con = null;
    public PlantillaCodigo unaPlantillaCodigo = new PlantillaCodigo();

    public PlantillaDAO(Connection unaCon) {
        con = unaCon;
    }

    public int buscarEstiloPlantilla(String unUsuario) throws Throwable {
        int valorRetornar = 0;

        String sql =
            "SELECT usuario,claveactual,plantillacss, codigoplantilla FROM usuarios where usuario = ?";
        PreparedStatement unPs = con.prepareStatement(sql);
        System.out.println("usuario: " + unUsuario);

        System.out.println("Llega a valida Usuario de PersonaDAO");
        unPs.setString(1, unUsuario);
        ResultSet unRs = unPs.executeQuery(); /*Creamos result set */
        if (unRs.next()) {
            valorRetornar = unRs.getInt("codigoplantilla");
            System.out.println("plantilla: " + valorRetornar);
        }
        unRs.close(); /* Cerramos result set xxxxxx*/
        unPs.close(); /* Cerramos Preapared statement */

        return valorRetornar;
```

```
    }

    public PlantillaCodigo recuperarPlantilla(int codigoPlantilla) throws Throwable {
        PlantillaCodigo unaPlantillaCodigo = new PlantillaCodigo();
        String sql = "SELECT * FROM plantilla where codigoplantilla = ?";
        PreparedStatement unPs = con.prepareStatement(sql);
        System.out.println("recuperarPlantilla");
        unPs.setInt(1, codigoPlantilla);
        ResultSet unRs = unPs.executeQuery();
        if (unRs.next()) {
            unaPlantillaCodigo.setCodigoBoton(unRs.getInt("codigoboton"));
            unaPlantillaCodigo.setCodigoEtiqueta(unRs.getInt("codigoetiqueta"));
            unaPlantillaCodigo.setCodigoTitulo(unRs.getInt("codigotitulo"));
            unaPlantillaCodigo.setCodigoSubtitulo(unRs.getInt("codigosubtitulo"));
            unaPlantillaCodigo.setCodigoFondoPagina(unRs.getInt("codigofondopagina"));
            unaPlantillaCodigo.setCodigoFondoFormulario(unRs.getInt("codigofondoformulario"));
            unaPlantillaCodigo.setCodigoCombo(unRs.getInt("codigocombo"));
            unaPlantillaCodigo.setCodigoLista(unRs.getInt("codigolista"));
            unaPlantillaCodigo.setCodigoImagenHeader(unRs.getInt("codigoimagenheader"));
            unaPlantillaCodigo.setCodigoImagenBody(unRs.getInt("codigoimagenbody"));
            unaPlantillaCodigo.setCodigoImagenFooter(unRs.getInt("codigoimagenfooter"));
unaPlantillaCodigo.setCodigoImagenFoto(unRs.getInt("codigoimagenfoto"));    }
        unRs.close();   unPs.close();    return unaPlantillaCodigo;    }

    public Plantilla buscarPlantilla(int codigoPlantilla) throws SQLException {
metodo que busca una plantilla guardada por el usuario que ingresó
        String sql1; String sql2; String sql3;  String sql4; String sql5;    String sql6;
        String sql7; String sql8; String sql9;  String sql10;   String sql11;
   Plantilla unaPlantilla = new Plantilla();
        sql1 =
"select tf.tipofuente as tipofuente, cf.rgb as colorfuente, tmf.tamaniofuente as tamaniofuente,
e.estilofuente as estilofuente\n" +
 "from tipofuente tf, colorfuente cf, tamaniofuente tmf, etiqueta e, plantilla p\n" +
 "where p.codigoplantilla = ? AND p.codigoetiqueta = e.codigoetiqueta and\n" +
 "e.codigotipofuente = tf.codigotipofuente AND e.codigocolorfuente = cf.codigocolorfuente\n" +
 "AND e.codigotamaniofuente=tmf.codigotamaniofuente";
        PreparedStatement unPs = con.prepareStatement(sql1);
        unPs.setInt(1, codigoPlantilla);
        ResultSet unRs = unPs.executeQuery();
Recibo los tipos, color, tamanio y estilo de fuente de una etiqueta
        if (unRs.next()) {
            unaPlantilla.setColorFuenteEtiqueta(unRs.getString("colorfuente"));
            unaPlantilla.setTamanioFuenteEtiqueta(unRs.getString("tamaniofuente"));
            unaPlantilla.setTipoFuenteEtiqueta(unRs.getString("tipofuente"));
            unaPlantilla.setCodigoPlantilla(codigoPlantilla);
            unaPlantilla.setEstiloFuenteEtiqueta(unRs.getString("estilofuente"));        }
        sql2 =
"select tf.tipofuente as tipofuente, cf.rgb as colorfuente, tmf.tamaniofuente as tamaniofuente,
t.estilofuente as estilofuente\n" +
 "from tipofuente tf, colorfuente cf, tamaniofuente tmf, titulo t, plantilla p\n" +
 "where p.codigoplantilla = ? AND p.codigotitulo = t.codigotitulo and\n" +
 "t.codigotipofuente = tf.codigotipofuente AND t.codigocolorfuente = cf.codigocolorfuente\n" +
 "AND t.codigotamaniofuente=tmf.codigotamaniofuente";
  unPs = con.prepareStatement(sql2);    unPs.setInt(1, codigoPlantilla);   unRs = unPs.executeQuery();
Recibo los tipos, color, tamanio y estilo de fuente de una titulo
if (unRs.next()) {
            unaPlantilla.setTipoFuenteTitulo(unRs.getString("tipofuente"));
            unaPlantilla.setColorFuenteTitulo(unRs.getString("colorfuente"));
            unaPlantilla.setTamanioFuenteTitulo(unRs.getString("tamaniofuente"));
```

```
        unaPlantilla.setEstiloFuenteTitulo(unRs.getString("estilofuente"));      }
    sql3 =
"select tf.tipofuente as tipofuente, cf.rgb as colorfuente, tmf.tamaniofuente as tamaniofuente, s.estilofuente as estilofuente\n" +
 "from tipofuente tf, colorfuente cf, tamaniofuente tmf, subtitulo s, plantilla p\n" +
 "where p.codigoplantilla = ? AND p.codigosubtitulo = s.codigosubtitulo and\n" +
 "s.codigotipofuente = tf.codigotipofuente AND s.codigocolorfuente = cf.codigocolorfuente\n" +
 "AND s.codigotamaniofuente=tmf.codigotamaniofuente";
    unPs = con.prepareStatement(sql3);
    unPs.setInt(1, codigoPlantilla);
    unRs = unPs.executeQuery();
```

Recibo los tipos, color, tamanio y estilo de fuente de una subtitulo

```
if (unRs.next()) {
unaPlantilla.setTipoFuenteSubtitulo(unRs.getString("tipofuente"));
unaPlantilla.setTamanioFuenteSubtitulo(unRs.getString("tamaniofuente"));
unaPlantilla.setColorFuenteSubtitulo(unRs.getString("colorfuente"));
unaPlantilla.setEstiloFuenteSubtitulo(unRs.getString("estilofuente"));  }
sql4 = "select cf.rgb as colorfondo from colorfondo cf, plantilla p, fondopagina fp\n" +
 "where p.codigofondopagina=fp.codigofondopagina and\n" +
 "fp.codigocolor=cf.codigocolor\n" + "and p.codigoplantilla=?";
unPs = con.prepareStatement(sql4);unPs.setInt(1, codigoPlantilla);unRs = unPs.executeQuery();
```

Recibo el color de fondo de pagina

```
if (unRs.next()) {
unaPlantilla.setColorFondoPagina(unRs.getString("colorfondo"));    }    sql5 =
"select tf.tipofuente as tipofuente, cf.rgb as colorfuente, cfd.descripcion as colorfondo\n" +  "from tipofuente tf, colorfuente cf, colorfondo cfd, combo c, plantilla p\n" +  "where p.codigoplantilla = ? AND p.codigocombo= c.codigocombo and\n" +  "c.codigotipofuente = tf.codigotipofuente AND c.codigocolorfuente = cf.codigocolorfuente\n" +  "AND c.codigocolorfondo=cfd.codigocolor";
unPs = con.prepareStatement(sql5);unPs.setInt(1, codigoPlantilla);unRs = unPs.executeQuery();
```

Recibo el color de fondo, color de fuente y tipo de fuente de un combo

```
if (unRs.next()) {
unaPlantilla.setColorFondoCombo(unRs.getString("colorfondo"));
unaPlantilla.setColorFuenteCombo(unRs.getString("colorfuente"));
unaPlantilla.setTipoFuenteCombo(unRs.getString("tipofuente"));     }
sql6 =
"select tf.tipofuente as tipofuente, cf.rgb as colorfuente, cfd.descripcion as colorfondo\n" +
 "from tipofuente tf, colorfuente cf, colorfondo cfd, listas l, plantilla p\n" +
 "where p.codigoplantilla = ? AND p.codigolista= l.codigolista and\n" +
 "l.codigotipofuente = tf.codigotipofuente AND l.codigocolorfuente = cf.codigocolorfuente\n" +
 "AND l.codigocolorfondo=cfd.codigocolor";
unPs = con.prepareStatement(sql6); unPs.setInt(1, codigoPlantilla);unRs = unPs.executeQuery();
```

Recibo el color de fondo, color de fuente y tipo de fuente de una lista

```
if (unRs.next()) {
unaPlantilla.setColorFondoLista(unRs.getString("colorfondo"));
unaPlantilla.setColorFuenteLista(unRs.getString("colorfuente"));
unaPlantilla.setTipoFuenteLista(unRs.getString("tipofuente"));    }
sql7 =
"select cf.rgb as colorfondo from colorfondo cf, plantilla p, fondoformulario ff\n" +
 "where p.codigofondoformulario=ff.codigofondoformulario and\n" +
 "ff.codigocolor=cf.codigocolor and p.codigoplantilla=?";
unPs = con.prepareStatement(sql7); unPs.setInt(1, codigoPlantilla);
unRs = unPs.executeQuery();
```

Recibo el color de fondo del formulario

```
if (unRs.next()) {
        unaPlantilla.setColorFondoFormulario(unRs.getString("colorfondo"));      }
```

```
sql8 = "select tf.tipofuente as tipofuente, cf.rgb as colorfuente, cfd.descripcion as colorfondo\n" +
 "from tipofuente tf, colorfuente cf, colorfondo cfd, boton b, plantilla p\n" +
 "where p.codigoplantilla = ? AND p.codigoboton= b.codigoboton and\n" +
 "b.codigotipofuente = tf.codigotipofuente AND b.codigocolorfuente = cf.codigocolorfuente\n" +
 "AND b.codigocolor=cfd.codigocolor";
unPs = con.prepareStatement(sql8);unPs.setInt(1, codigoPlantilla);unRs = unPs.executeQuery();
```

Recibo los datos de boton

```
if (unRs.next()) {
unaPlantilla.setColorFuenteBoton(unRs.getString("colorfuente"));
unaPlantilla.setColorFondoBoton(unRs.getString("colorfondo"));
unaPlantilla.setTipoFuenteBoton(unRs.getString("tipofuente"));     }
sql9 = "select i.descripcion as imagenheader\n" + "from imagen i, plantilla p\n" +
 "where p.codigoimagenheader =i.codigoimagen and p.codigoplantilla=?";
unPs = con.prepareStatement(sql9); unPs.setInt(1, codigoPlantilla);
unRs = unPs.executeQuery();
```

Recibo los datos de boton

```
if (unRs.next()) {
unaPlantilla.setImagenHeader(unRs.getString("imagenheader"));     }
sql10 =  "select i.descripcion as imagenbody\n" + "from imagen i, plantilla p\n" +
"where p.codigoimagenbody =i.codigoimagen and p.codigoplantilla=?";
unPs = con.prepareStatement(sql10);   unPs.setInt(1, codigoPlantilla);
unRs = unPs.executeQuery();  if (unRs.next()) {
unaPlantilla.setImagenBody(unRs.getString("imagenbody"));     }
sql11 =  "select i.descripcion as imagenfooter\n" + "from imagen i, plantilla p\n" +
"where p.codigoimagenfooter =i.codigoimagen and p.codigoplantilla=?";
unPs = con.prepareStatement(sql11); unPs.setInt(1, codigoPlantilla);
unRs = unPs.executeQuery();    if (unRs.next()) {
unaPlantilla.setImagenFooter(unRs.getString("imagenfooter"));     }
```

Buscar el codigo de la imagen foto

```
sql1 = "select codigoimagenfoto from plantilla where codigoplantilla =?";
unPs = con.prepareStatement(sql1); unPs.setInt(1, codigoPlantilla);
unRs = unPs.executeQuery();if (unRs.next()) {
unaPlantilla.setImagenFoto(unRs.getInt("codigoimagenfoto"));     }
unRs.close(); unPs.close(); return unaPlantilla; }

public int buscarCodigo() throws Throwable {  int codigo = 0;
String sql = "select max(codigoplantilla) as codigoplantilla from plantilla";
PreparedStatement unPs = con.prepareStatement(sql);
ResultSet unRs = unPs.executeQuery();while (unRs.next()) {codigo = unRs.getInt("codigoplantilla");    }
unRs.close(); unPs.close();  return codigo; }
public boolean guardar(PlantillaCodigo unaPlantillaNueva) throws Throwable { String sql =
"INSERT INTO plantilla (codigoplantilla, codigoetiqueta,codigotitulo, \n" +
"codigosubtitulo,codigoboton,codigofondopagina,codigofondoformulario," +
"codigocombo,codigolista,codigoimagenheader,codigoimagenbody,codigoimagenfooter)" + "values
(?,?,?,?,?,?,?,?,?,?,?,?)";PreparedStatement unPs = con.prepareStatement(sql); unPs.setInt(1,
unaPlantillaNueva.getCodigoPlantilla());
unPs.setInt(2, unaPlantillaNueva.getCodigoEtiqueta());unPs.setInt(3,
unaPlantillaNueva.getCodigoTitulo());unPs.setInt(4, unaPlantillaNueva.getCodigoSubtitulo());
unPs.setInt(5, unaPlantillaNueva.getCodigoBoton());
unPs.setInt(6, unaPlantillaNueva.getCodigoFondoPagina());
unPs.setInt(7, unaPlantillaNueva.getCodigoFondoFormulario());
unPs.setInt(8, unaPlantillaNueva.getCodigoCombo());
unPs.setInt(9, unaPlantillaNueva.getCodigoLista());
unPs.setInt(10, unaPlantillaNueva.getCodigoImagenHeader());
unPs.setInt(11, unaPlantillaNueva.getCodigoImagenBody());
unPs.setInt(12, unaPlantillaNueva.getCodigoImagenFooter());
```

int i = unPs.executeUpdate(); unPs.close(); if (i == 1) return true; else return false;
} public int buscarCodigoPlantilla(String unUsuario) throws Throwable { int codigo = 0; String sql = "SELECT usuario,claveactual, codigoplantilla FROM usuarios where usuario = ?"; PreparedStatement unPs = con.prepareStatement(sql); System.out.println("usuario: " + unUsuario); System.out.println("Llega a buscarCodigoPlantilla de PersonaDAO"); unPs.setString(1, unUsuario); ResultSet unRs = unPs.executeQuery();

Creamos result set
if (unRs.next()) {codigo = unRs.getInt("codigoplantilla");
System.out.println("plantilla: " + codigo); }
unRs.close();
Cerramos result set
unPs.close(); **Cerramos Preapared statement**
return codigo; }

Se utiliza para actualizar los datos de la plantilla se usa en la clase ActualizarPlantillaAction
public boolean actualizar(PlantillaCodigo unaPlantillaNueva) throws Throwable {
String sql =
"UPDATE plantilla SET codigoetiqueta =?," + "" + " codigotitulo = ?," +
" codigosubtitulo=?," +
" codigoboton=?," +
" codigofondopagina=?, " +
" codigofondoformulario=?," +
" codigocombo=?," +
" codigolista=?," +
" codigoimagenfoto=?" +
" WHERE codigoplantilla =?";
PreparedStatement unPs = con.prepareStatement(sql);unPs.setInt(1, unaPlantillaNueva.getCodigoEtiqueta());unPs.setInt(2, unaPlantillaNueva.getCodigoTitulo());unPs.setInt(3, unaPlantillaNueva.getCodigoSubtitulo());unPs.setInt(4, unaPlantillaNueva.getCodigoBoton());unPs.setInt(5, unaPlantillaNueva.getCodigoFondoPagina());unPs.setInt(6, unaPlantillaNueva.getCodigoFondoFormulario());unPs.setInt(7, unaPlantillaNueva.getCodigoCombo());unPs.setInt(8, unaPlantillaNueva.getCodigoLista());unPs.setInt(9, unaPlantillaNueva.getCodigoImagenFoto());unPs.setInt(10, unaPlantillaNueva.getCodigoPlantilla()); int i = unPs.executeUpdate();
unPs.close(); if (i == 1) return true; else return false; }

Para traer todas las plantillas predeterminadas que tienen un nombre
public Vector buscarTodos() throws Throwable {
Vector valorDevolver = new Vector();
String sql =
"select codigoplantilla, nombre from plantilla\n" + "where nombre != "or nombre != null;";
PreparedStatement unPs = con.prepareStatement(sql);
System.out.println("Llega a buscartodos de PlantillaDAO");
ResultSet unRs = unPs.executeQuery(); PlantillaCodigo unaPlantillaCodigo;
while (unRs.next()) { unaPlantillaCodigo = new PlantillaCodigo();
unaPlantillaCodigo.setCodigoPlantilla(unRs.getInt("codigoplantilla"));
unaPlantillaCodigo.setNombre(unRs.getString("nombre"));
valorDevolver.addElement(unaPlantillaCodigo); }
unRs.close(); unPs.close(); return valorDevolver; }

Método que busca una plantilla guardada por el usuario
public Plantilla recuperaCodigo(int codigoPlantilla) throws SQLException
String sql1; String sql2; String sql3; String sql4; String sql5; String sql6; String sql7; String sql8; String sql9; String sql10; String sql11; Plantilla unaPlantilla = new Plantilla();

```
sql1 = "select tf.codigotipofuente as tipofuente, cf.codigocolorfuente as colorfuente, tmf.codigotamaniofuente as tamaniofuente, e.estilofuente as estilofuente\n" +
"from tipofuente tf, colorfuente cf, tamaniofuente tmf, etiqueta e, plantilla p\n" +
"where p.codigoplantilla = ? AND p.codigoetiqueta = e.codigoetiqueta and\n" +
"e.codigotipofuente = tf.codigotipofuente AND e.codigocolorfuente = cf.codigocolorfuente\n" + "AND e.codigotamaniofuente=tmf.codigotamaniofuente";
PreparedStatement unPs = con.prepareStatement(sql1);unPs.setInt(1, codigoPlantilla); ResultSet unRs = unPs.executeQuery();
```

Recibo los tipos, color, tamanio y estilo de fuente de una etiqueta

```
if (unRs.next()) {
unaPlantilla.setColorFuenteEtiqueta(unRs.getString("colorfuente"));
unaPlantilla.setTamanioFuenteEtiqueta(unRs.getString("tamaniofuente"));
unaPlantilla.setTipoFuenteEtiqueta(unRs.getString("tipofuente"));
unaPlantilla.setCodigoPlantilla(codigoPlantilla);
unaPlantilla.setEstiloFuenteEtiqueta(unRs.getString("estilofuente"));    }
sql2 =  "select tf.codigotipofuente as tipofuente, cf.codigocolorfuente as colorfuente, tmf.codigotamaniofuente as tamaniofuente, t.estilofuente as estilofuente\n" +
"from tipofuente tf, colorfuente cf, tamaniofuente tmf, titulo t, plantilla p\n" +
"where p.codigoplantilla = ? AND p.codigotitulo = t.codigotitulo and\n" +
"t.codigotipofuente = tf.codigotipofuente AND t.codigocolorfuente = cf.codigocolorfuente\n" +
"AND t.codigotamaniofuente=tmf.codigotamaniofuente";
unPs = con.prepareStatement(sql2);  unPs.setInt(1, codigoPlantilla); unRs = unPs.executeQuery();
```

Recibo los tipos, color, tamanio y estilo de fuente de un titulo

```
if (unRs.next()) {
unaPlantilla.setTipoFuenteTitulo(unRs.getString("tipofuente"));
unaPlantilla.setColorFuenteTitulo(unRs.getString("colorfuente"));
unaPlantilla.setTamanioFuenteTitulo(unRs.getString("tamaniofuente"));
unaPlantilla.setEstiloFuenteTitulo(unRs.getString("estilofuente"));    }
sql3 = "select tf.codigotipofuente as tipofuente, cf.codigocolorfuente as colorfuente, tmf.codigotamaniofuente as tamaniofuente, s.estilofuente as estilofuente\n" +
"from tipofuente tf, colorfuente cf, tamaniofuente tmf, subtitulo s, plantilla p\n" +
"where p.codigoplantilla = ? AND p.codigosubtitulo = s.codigosubtitulo and\n" +
"s.codigotipofuente = tf.codigotipofuente AND s.codigocolorfuente = cf.codigocolorfuente\n" + "AND s.codigotamaniofuente=tmf.codigotamaniofuente";
unPs = con.prepareStatement(sql3); unPs.setInt(1, codigoPlantilla);unRs = unPs.executeQuery();
```

Recibo los tipos, color, tamanio y estilo de fuente de una subtitulo

```
if (unRs.next()) {
unaPlantilla.setTipoFuenteSubtitulo(unRs.getString("tipofuente"));
unaPlantilla.setTamanioFuenteSubtitulo(unRs.getString("tamaniofuente"));
unaPlantilla.setColorFuenteSubtitulo(unRs.getString("colorfuente"));
unaPlantilla.setEstiloFuenteSubtitulo(unRs.getString("estilofuente"));   }
sql4 = "select cf.codigocolor as colorfondo from colorfondo cf, plantilla p, fondopagina fp\n" +  "where p.codigofondopagina=fp.codigofondopagina and\n" +
"fp.codigocolor=cf.codigocolor\n" + "and p.codigoplantilla=?";
unPs = con.prepareStatement(sql4); unPs.setInt(1, codigoPlantilla);unRs = unPs.executeQuery();
```

Recibo el color de fondo de pagina

```
if (unRs.next()) {
unaPlantilla.setColorFondoPagina(unRs.getString("colorfondo"));  }
sql5 =  "select tf.codigotipofuente as tipofuente, cf.codigocolorfuente as colorfuente, cfd.codigocolor as colorfondo\n" +  "from tipofuente tf, colorfuente cf, colorfondo cfd, combo c, plantilla p\n" + "where p.codigoplantilla = ? AND p.codigocombo= c.codigocombo and\n" + "c.codigotipofuente = tf.codigotipofuente AND c.codigocolorfuente = cf.codigocolorfuente\n" +"AND c.codigocolorfondo=cfd.codigocolor";   unPs = con.prepareStatement(sql5);
unPs.setInt(1, codigoPlantilla); unRs = unPs.executeQuery();
```

Recibo el color de fondo, color de fuente y tipo de fuente de un combo

```
if (unRs.next()) {
unaPlantilla.setColorFondoCombo(unRs.getString("colorfondo"));
unaPlantilla.setColorFuenteCombo(unRs.getString("colorfuente"));
unaPlantilla.setTipoFuenteCombo(unRs.getString("tipofuente")); }
sql6 =  "select tf.codigotipofuente as tipofuente, cf.codigocolorfuente as colorfuente, cfd.codigocolor as colorfondo\n" +  "from tipofuente tf, colorfuente cf, colorfondo cfd, listas l, plantilla p\n" + "where p.codigoplantilla = ? AND p.codigolista= l.codigolista and\n" +"l.codigotipofuente = tf.codigotipofuente AND l.codigocolorfuente = cf.codigocolorfuente\n" + "AND l.codigocolorfondo=cfd.codigocolor";unPs = con.prepareStatement(sql6); unPs.setInt(1, codigoPlantilla);unRs = unPs.executeQuery();
```

Recibo el color de fondo, color de fuente y tipo de fuente de una lista

```
if (unRs.next()) {
unaPlantilla.setColorFondoLista(unRs.getString("colorfondo"));
unaPlantilla.setColorFuenteLista(unRs.getString("colorfuente"));
unaPlantilla.setTipoFuenteLista(unRs.getString("tipofuente"));  }
sql7 = "select cf.codigocolor as colorfondo from colorfondo cf, plantilla p, fondoformulario ff\n" +"where p.codigofondoformulario=ff.codigofondoformulario and\n" +  "ff.codigocolor=cf.codigocolor and p.codigoplantilla=?";
unPs = con.prepareStatement(sql7); unPs.setInt(1, codigoPlantilla);
unRs = unPs.executeQuery();
```

Recibo el color de fondo del formulario

```
if (unRs.next()) {
unaPlantilla.setColorFondoFormulario(unRs.getString("colorfondo"));    }
sql8 =  "select tf.codigotipofuente as tipofuente, cf.codigocolorfuente as colorfuente, cfd.codigocolor as colorfondo\n" +  "from tipofuente tf, colorfuente cf, colorfondo cfd, boton b, plantilla p\n" + "where p.codigoplantilla = ? AND p.codigoboton= b.codigoboton and\n" +"b.codigotipofuente = tf.codigotipofuente AND b.codigocolorfuente = cf.codigocolorfuente\n" +"AND b.codigocolor=cfd.codigocolor";
unPs = con.prepareStatement(sql8);  unPs.setInt(1, codigoPlantilla);unRs = unPs.executeQuery();
```

Recibo los datos de boton

```
if (unRs.next()) {
unaPlantilla.setColorFuenteBoton(unRs.getString("colorfuente"));
unaPlantilla.setColorFondoBoton(unRs.getString("colorfondo"));
unaPlantilla.setTipoFuenteBoton(unRs.getString("tipofuente"));       }
sql9 = "select i.descripcion as imagenheader\n" + "from imagen i, plantilla p\n" +
"where p.codigoimagenheader =i.codigoimagen and p.codigoplantilla=?";
unPs = con.prepareStatement(sql9); unPs.setInt(1, codigoPlantilla);unRs = unPs.executeQuery();
```

Recibo los datos de boton

```
if (unRs.next()) {
unaPlantilla.setImagenHeader(unRs.getString("imagenheader"));     }   sql10 =
"select i.descripcion as imagenbody\n" + "from imagen i, plantilla p\n" +
"where p.codigoimagenbody =i.codigoimagen and p.codigoplantilla=?"; unPs = con.prepareStatement(sql10); unPs.setInt(1, codigoPlantilla);unRs = unPs.executeQuery(); if (unRs.next()) {
unaPlantilla.setImagenBody(unRs.getString("imagenbody"));     }    sql11 =
"select i.descripcion as imagenfooter\n" + "from imagen i, plantilla p\n" +
"where p.codigoimagenfooter =i.codigoimagen and p.codigoplantilla=?";
unPs = con.prepareStatement(sql11);  unPs.setInt(1, codigoPlantilla); unRs = unPs.executeQuery();  if (unRs.next()) {
unaPlantilla.setImagenFooter(unRs.getString("imagenfooter"));    }
```

Buscar el codigo de la imagen foto

```
sql1 = "select codigoimagenfoto from plantilla where codigoplantilla =?"; unPs = con.prepareStatement(sql1); unPs.setInt(1, codigoPlantilla); unRs = unPs.executeQuery(); if (unRs.next()) {
unaPlantilla.setImagenFoto(unRs.getInt("codigoimagenfoto"));    }
unRs.close();  unPs.close(); return unaPlantilla; }}
```

SubtituloDAO.java

Descripción: Se declara las sentencias SQL a la base de datos, mediante dos metodos, uno recupera el codigo del subtitulo BuscarCodigo, y el otro guarda el Subtitulo en Guardar.

Bean: Subtitulo.

```
package com.bvg.dao;
import com.bvg.bean.Subtitulo;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
public class SubtituloDAO {   private Connection con = null;
public SubtituloDAO(Connection unaCon) {    con = unaCon;    }
  public int buscarCodigo() throws Throwable {   int codigo = 0; String sql = "SELECT codigosubtitulo
FROM subtitulo";    PreparedStatement unPs = con.prepareStatement(sql); ResultSet unRs =
unPs.executeQuery();  while (unRs.next())
 { if (unRs.isLast()) {codigo = unRs.getInt("codigosubtitulo");      }
    }  unRs.close(); unPs.close(); return codigo;    } public boolean guardar(Subtitulo unSubtitulo) throws
Throwable { String sql =
  "INSERT INTO subtitulo (codigosubtitulo,codigotipofuente, codigocolorfuente, codigotamaniofuente)
values (?,?,?,?)";PreparedStatement unPs = con.prepareStatement(sql);
 unPs.setInt(1, unSubtitulo.getCodigoSubtitulo());
 unPs.setString(2, unSubtitulo.getCodigoTipoFuente());
 unPs.setString(3,                                         unSubtitulo.getCodigoColorFuente());unPs.setString(4,
unSubtitulo.getCodigoTamanioFuente());  int i = unPs.executeUpdate();
      unPs.close(); if (i == 1)  return true;   else    return false; }
 public int compararDatosSubtitulo(String tipoFuenteSubtitulo,  String colorFuenteSubtitulo,
       String tamanioFuenteSubtitulo) throws Throwable {  int codigo = 0;  String sql =
   "select codigosubtitulo from subtitulo where codigotipofuente = ? And" +    " codigocolorfuente = ? and
codigotamaniofuente = ?";
 PreparedStatement unPs = con.prepareStatement(sql);  unPs.setString(1, tipoFuenteSubtitulo);
   unPs.setString(2, colorFuenteSubtitulo);  unPs.setString(3, tamanioFuenteSubtitulo);
ResultSet unRs = unPs.executeQuery(); while (unRs.next()) {    if (unRs.isFirst()) {
codigo = unRs.getInt("codigosubtitulo");   } }unRs.close(); unPs.close();return codigo;   }}
```

TamanioFuenteDAO.java

Descripción: Se declara las sentencias SQL a la base de datos, mediante un vector que busca el tamaño de la fuente.

Bean: TamanioFuente.

```
package com.bvg.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Vector;
import java.lang.String;
import com.bvg.bean.TamanioFuente;
public class TamanioFuenteDAO { private Connection con = null; public
TamanioFuenteDAO(Connection unaCon) { con = unaCon; }
public Vector buscarTodos() throws Throwable { Vector valorDevolver = new Vector(); String sql =
"SELECT * FROM tamaniofuente"; PreparedStatement unPs = con.prepareStatement(sql); ResultSet
unRs = unPs.executeQuery(); TamanioFuente unTamanioFuente; while (unRs.next()) {
unTamanioFuente = new
TamanioFuente();unTamanioFuente.setCodigoTamanioFuente(unRs.getString("codigotamaniofuente"))
;unTamanioFuente.setTamnioFuente(unRs.getString("tamaniofuente"));
valorDevolver.addElement(unTamanioFuente); }unRs.close(); unPs.close(); return valorDevolver; }}
```

TipoFuenteDAO.java

Descripción: Se declara las sentencias SQL a la base de datos, mediante un vector que busca el tipo de la fuente BuscarTodos.

Bean: TipoFuente.

```
package com.bvg.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Vector;
import java.lang.String; import com.bvg.bean.TipoFuente;
public class TipoFuenteDAO {
 private Connection con = null;
public TipoFuenteDAO(Connection unaCon) {
 con = unaCon; } public Vector buscarTodos() throws Throwable {
Vector valorDevolver = new Vector(); String sql = "SELECT * FROM tipofuente ORDER BY
tipofuente";
PreparedStatement unPs = con.prepareStatement(sql);
 ResultSet unRs = unPs.executeQuery();
 TipoFuente untipoFuente; while (unRs.next()) { untipoFuente = new TipoFuente();
untipoFuente.setCodigoTipoFuente(unRs.getString("codigotipofuente"));untipoFuente.setTipoFuente(un
Rs.getString("tipofuente"));valorDevolver.addElement(untipoFuente); }
unRs.close(); unPs.close(); return valorDevolver; }}
```

TituloDAO.java

Descripción: Se declara las sentencias SQL a la base de datos, mediante dos metodos, uno busca el codigo del titulo BuscarCodigo, y el otro guarda el titulo en Guardar.

Bean: Titulo.

```
package com.bvg.dao;
import com.bvg.bean.Titulo;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
public class TituloDAO {   private Connection con = null; public TituloDAO(Connection unaCon) {  con =
unaCon; }   public int buscarCodigo() throws Throwable {   int codigo = 0;    String sql = "SELECT
codigotitulo FROM titulo";         PreparedStatement unPs = con.prepareStatement(sql);       ResultSet
unRs = unPs.executeQuery();        while (unRs.next()) {          if (unRs.isLast()) {codigo =
unRs.getInt("codigotitulo");   }   } unRs.close();  unPs.close();  return codigo;  }
public boolean guardar(Titulo unTitulo) throws Throwable {  String sql =           "INSERT INTO titulo
(codigotitulo,codigotipofuente,     codigocolorfuente,     codigotamaniofuente)     values     (?,?,?,?)";
PreparedStatement unPs = con.prepareStatement(sql);   unPs.setInt(1, unTitulo.getCodigoTitulo());
unPs.setString(2,                    unTitulo.getCodigoTipoFuente());                       unPs.setString(3,
unTitulo.getCodigoColorFuente());unPs.setString(4,  unTitulo.getCodigoTamanioFuente());      int  i  =
unPs.executeUpdate();      unPs.close();  if (i == 1) return true;  else return false; }
public int compararDatosTitulo(String tipoFuenteTitulo, String colorFuenteTitulo,
String tamanioFuenteTitulo) throws Throwable {  int codigo = 0; String sql =           "select codigotitulo
from titulo where codigotipofuente = ? And" +  " codigocolorfuente = ? and codigotamaniofuente = ?";
PreparedStatement unPs = con.prepareStatement(sql);  unPs.setString(1, tipoFuenteTitulo);
unPs.setString(2, colorFuenteTitulo);  unPs.setString(3, tamanioFuenteTitulo);        ResultSet unRs =
unPs.executeQuery();  while  (unRs.next())  {    if  (unRs.isFirst())  {                        codigo  =
unRs.getInt("codigotitulo");    }   } unRs.close(); unPs.close(); return codigo; }}
```

Formulario

EstiloFormulario.java

Descripción: Se declara los campos que va a ser usado por el usuario, el tipo de fuente de la etiqueta, el color de fuente de la etiqueta, el tamaño de fuente de la etiqueta, el tipo de fuente del titulo, el color de fuente del titulo, el

tamaño de fuente del titulo, el tipo de fuente del subtitulo, el color de fuente del subtitulo, el tamaño de fuente del subtitulo, el color de fondo de pagina, el color del borde, el color del fondo del formulario, la imagen de fondo de la pagina, la imagen de la cabecera, el tipo de fuente del combo, el color de fuente del combo, el color del fondo del combo, el tipo de fuente del select, el color de fuente del select, el color del fondo del select.

Métodos: Get y Set.

package com.bvg.formulario;

```
import org.apache.struts.action.ActionForm;
import org.apache.struts.upload.FormFile;
public class EstiloFormulario extends ActionForm {
private int codigoPlantilla;
private String tipoFuenteEtiqueta;
private String colorFuenteEtiqueta;
private String tamanioFuenteEtiqueta;
private String tipoFuenteTitulo;
private String colorFuenteTitulo;
private String tamanioFuenteTitulo;
private String tipoFuenteSubtitulo;
private String colorFuenteSubtitulo;
private String tamanioFuenteSubtitulo;
private String usuario;
private String colorFondoPagina;
private String colorBorde;
private String colorFondoFormulario;
private FormFile imagenFondoPagina;
private FormFile imagenCabecera;
private String tipoFuenteCombo;
private String colorFuenteCombo;
private String colorFondoCombo;
private String tipoFuenteLista;
private String colorFuenteLista;
private String colorFondoLista;
private String tipoFuenteBoton;
private String colorFuenteBoton;
private String colorFondoBoton;
private String estadoEtiquetacolor="0";
private String estadoTitulocolor="0";
private String estadoSubtitulocolor ="0";
private String estadoEtiquetafuente ="0";
private String estadoTitulofuente ="0";
private String estadoSubtitulofuente ="0";
private String estadoEtiquetatamanio ="0";
```

```
private String estadoTitulotamanio="0";
private String estadoSubtitulotamanio ="0";
private String estadoListacolor ="0";
private String estadoListafuente ="0";
private String estadoListatamanio ="0";
private String estadoCombocolor="0";
private String estadoCombofuente ="0";
private String estadoCombotamanio ="0";
private String estadoColorformulario ="0";
private String estadoBotonfuente ="0";
private String estadoBotonColorFuente="0";
private String estadoBotontipofuente ="0";
private String estadoColorPagina="0";
private String estadoColorBorde="0";
private String estadoImagenCabecera ="0";
public EstiloFormulario() {  }
public void setTipoFuenteEtiqueta(String tipoFuenteEtiqueta) {
this.tipoFuenteEtiqueta = tipoFuenteEtiqueta;  }
public String getTipoFuenteEtiqueta() {   return tipoFuenteEtiqueta;
}public void setColorFuenteEtiqueta(String colorFuenteEtiqueta) {
 this.colorFuenteEtiqueta = colorFuenteEtiqueta;  }
public String getColorFuenteEtiqueta() {    return colorFuenteEtiqueta;    }
public void setTamanioFuenteEtiqueta(String tamanioFuenteEtiqueta) {
this.tamanioFuenteEtiqueta = tamanioFuenteEtiqueta;  }
public String getTamanioFuenteEtiqueta() { return tamanioFuenteEtiqueta;  }
public void setTipoFuenteTitulo(String tipoFuenteTitulo) {   this.tipoFuenteTitulo = tipoFuenteTitulo;   }
public String getTipoFuenteTitulo() { return tipoFuenteTitulo;  }
public void setColorFuenteTitulo(String colorFuenteTitulo) { this.colorFuenteTitulo =
colorFuenteTitulo; }
public String getColorFuenteTitulo() {  return colorFuenteTitulo; }
public void setTamanioFuenteTitulo(String tamanioFuenteTitulo) {
this.tamanioFuenteTitulo = tamanioFuenteTitulo;    }
public String getTamanioFuenteTitulo() {return tamanioFuenteTitulo; }
public void setTipoFuenteSubtitulo(String tipoFuenteSubtitulo) { this.tipoFuenteSubtitulo =
tipoFuenteSubtitulo;  }
public String getTipoFuenteSubtitulo() {  return tipoFuenteSubtitulo;  }
public void setColorFuenteSubtitulo(String colorFuenteSubtitulo) {
this.colorFuenteSubtitulo = colorFuenteSubtitulo; }
public String getColorFuenteSubtitulo() { return colorFuenteSubtitulo;  }
public void setTamanioFuenteSubtitulo(String tamanioFuenteSubtitulo) {
this.tamanioFuenteSubtitulo = tamanioFuenteSubtitulo;    }
public String getTamanioFuenteSubtitulo() {  return tamanioFuenteSubtitulo;  }
public void setUsuario(String usuario) {    this.usuario = usuario;  }
public String getUsuario() {   return usuario; }
public void setColorFondoPagina(String colorFondoPagina) { this.colorFondoPagina =
colorFondoPagina; }
public String getColorFondoPagina() {   return colorFondoPagina; }
public void setColorFondoFormulario(String colorFondoFormulario) {   this.colorFondoFormulario =
colorFondoFormulario;  }
public String getColorFondoFormulario() {    return colorFondoFormulario;   }
public void setColorBorde(String colorBorde) {    this.colorBorde = colorBorde;   }
public String getColorBorde() {   return colorBorde; }
public void setTipoFuenteCombo(String tipoFuenteCombo) { this.tipoFuenteCombo =
tipoFuenteCombo;   }
public String getTipoFuenteCombo() {     return tipoFuenteCombo;  }
public void setColorFuenteCombo(String colorFuenteCombo) { this.colorFuenteCombo =
colorFuenteCombo;  }
public String getColorFuenteCombo() {   return colorFuenteCombo;   }
```

```
public void setColorFondoCombo(String colorFondoCombo) { this.colorFondoCombo = colorFondoCombo; }
public String getColorFondoCombo() { return colorFondoCombo; }
public void setCodigoPlantilla(int codigoPlantilla) {this.codigoPlantilla = codigoPlantilla; }
public int getCodigoPlantilla() { return codigoPlantilla; }
public void setTipoFuenteLista(String tipoFuenteLista) {this.tipoFuenteLista = tipoFuenteLista; }
public String getTipoFuenteLista() { return tipoFuenteLista; }
public void setColorFuenteLista(String colorFuenteLista) { this.colorFuenteLista = colorFuenteLista; }
public String getColorFuenteLista() { return colorFuenteLista; } public void setColorFondoLista(String colorFondoLista) { this.colorFondoLista = colorFondoLista; } public String getColorFondoLista() { return colorFondoLista; }
public void setTipoFuenteBoton(String tipoFuenteBoton) { this.tipoFuenteBoton = tipoFuenteBoton; }
public String getTipoFuenteBoton() { return tipoFuenteBoton; } public void setColorFuenteBoton(String colorFuenteBoton) { this.colorFuenteBoton = colorFuenteBoton; }
public String getColorFuenteBoton() {
return colorFuenteBoton; } public void setColorFondoBoton(String colorFondoBoton) { this.colorFondoBoton = colorFondoBoton; } public String getColorFondoBoton() { return colorFondoBoton; } public FormFile getImagenFondoPagina() { return imagenFondoPagina; }
public void setImagenFondoPagina(FormFile imagenFondoPagina) { this.imagenFondoPagina = imagenFondoPagina; }
public FormFile getImagenCabecera() { return imagenCabecera; } public void setImagenCabecera(FormFile imagenCabecera) { this.imagenCabecera = imagenCabecera; } public void setEstadoEtiquetacolor(String estadoEtiquetacolor) {
this.estadoEtiquetacolor = estadoEtiquetacolor; }public String getEstadoEtiquetacolor() { return estadoEtiquetacolor; }public void setEstadoTitulocolor(String estadoTitulocolor) {this.estadoTitulocolor = estadoTitulocolor; } public String getEstadoTitulocolor() { return estadoTitulocolor; } public void setEstadoSubtitulocolor(String estadoSubtitulocolor) { this.estadoSubtitulocolor = estadoSubtitulocolor; } public String getEstadoSubtitulocolor() { return estadoSubtitulocolor; } public void setEstadoEtiquetafuente(String estadoEtiquetafuente) { this.estadoEtiquetafuente = estadoEtiquetafuente; } public String getEstadoEtiquetafuente() {return estadoEtiquetafuente; } public void setEstadoTitulofuente(String estadoTitulofuente) { this.estadoTitulofuente = estadoTitulofuente; }public String getEstadoTitulofuente() {return estadoTitulofuente; }public void setEstadoSubtitulofuente(String estadoSubtitulofuente) { this.estadoSubtitulofuente = estadoSubtitulofuente; } public String getEstadoSubtitulofuente() { return estadoSubtitulofuente; }
public void setEstadoEtiquetatamanio(String estadoEtiquetatamanio) { this.estadoEtiquetatamanio = estadoEtiquetatamanio; } public String getEstadoEtiquetatamanio() { return estadoEtiquetatamanio; } public void setEstadoTitulotamanio(String estadoTitulotamanio) { this.estadoTitulotamanio = estadoTitulotamanio; } public String getEstadoTitulotamanio() { return estadoTitulotamanio; }public void setEstadoSubtitulotamanio(String estadoSubtitulotamanio) { this.estadoSubtitulotamanio = estadoSubtitulotamanio; }
public String getEstadoSubtitulotamanio() { return estadoSubtitulotamanio; } public void setEstadoListacolor(String estadoListacolor) {this.estadoListacolor = estadoListacolor; } public String getEstadoListacolor() { return estadoListacolor;
} public void setEstadoListafuente(String estadoListafuente) { this.estadoListafuente = estadoListafuente; } public String getEstadoListafuente() {
return estadoListafuente; }public void setEstadoListatamanio(String estadoListatamanio) { this.estadoListatamanio = estadoListatamanio; } public String getEstadoListatamanio() { return estadoListatamanio; } public void setEstadoCombocolor(String estadoCombocolor) { this.estadoCombocolor = estadoCombocolor; } public String getEstadoCombocolor() { return estadoCombocolor; }public void setEstadoCombofuente(String estadoCombofuente) { this.estadoCombofuente = estadoCombofuente; }public String getEstadoCombofuente() { return estadoCombofuente; } public void setEstadoCombotamanio(String estadoCombotamanio) { this.estadoCombotamanio = estadoCombotamanio; }public String getEstadoCombotamanio() { return estadoCombotamanio; } public void setEstadoColorformulario(String estadoColorformulario) { this.estadoColorformulario = estadoColorformulario; } public String getEstadoColorformulario() { return estadoColorformulario; } public void setEstadoBotonfuente(String estadoBotonfuente) {this.estadoBotonfuente = estadoBotonfuente; } public String getEstadoBotonfuente() {return estadoBotonfuente; }public void setEstadoBotonColorFuente(String estadoBotonColorFuente)
```

```
{this.estadoBotonColorFuente = estadoBotonColorFuente; } public String
getEstadoBotonColorFuente() { return estadoBotonColorFuente;
 } public void setEstadoBotontipofuente(String estadoBotontipofuente) {
   this.estadoBotontipofuente = estadoBotontipofuente; } public String getEstadoBotontipofuente() {
 return estadoBotontipofuente; } public void setEstadoColorPagina(String estadoColorPagina) {
  this.estadoColorPagina = estadoColorPagina; }
 public String getEstadoColorPagina() {return estadoColorPagina; }
  public void setEstadoColorBorde(String estadoColorBorde) {
this.estadoColorBorde = estadoColorBorde; } public String getEstadoColorBorde() {
return estadoColorBorde; }public void setEstadoImagenCabecera(String estadoImagenCabecera) {
this.estadoImagenCabecera = estadoImagenCabecera; }
 public String getEstadoImagenCabecera() {return estadoImagenCabecera; }}
```

PersonaForm.java

Descripción: Se declara los campos que va a ser usado por el usuario, su usuario, su clave, el nombre, el apellido, y su plantilla.

Métodos: Get y Set.

```
package com.bvg.formulario;

import org.apache.struts.action.ActionForm;
public class PersonaForm extends ActionForm {
private String nombre = ""; private String apellido = "";
private String usuario = "";
 private String clave = ""; private String plantilla = "";
public void setNombre(String nombre) {
 this.nombre = nombre; }
public String getNombre() {
 return nombre; }
public void setApellido(String apellido) {
  this.apellido = apellido; }
public String getApellido() {
   return apellido; }
public String getUsuario() {
   return usuario; }
public void setUsuario(String usuario) {
    this.usuario = usuario; }
public String getClave() { return clave; }
public void setClave(String clave) {
   this.clave = clave; }
public void setPlantilla(String plantilla) {
this.plantilla = plantilla; }
public String getPlantilla() {
 return plantilla; }}
```

Servlets

EstiloFormulario.java

Descripción: Clase por la cual se recupera el usuario y el estilo de plantilla escogida e invoca como respuesta el Input.jsp.

Enlaces: PlantillaBO.

```
package com.bvg.formulario;
import org.apache.struts.action.ActionForm;
import org.apache.struts.upload.FormFile;
public class EstiloFormulario extends ActionForm {
private int codigoPlantilla;  private String tipoFuenteEtiqueta;
private String colorFuenteEtiqueta;   private String tamanioFuenteEtiqueta;
private String tipoFuenteTitulo;    private String colorFuenteTitulo;
private String tamanioFuenteTitulo;    private String tipoFuenteSubtitulo;
private String colorFuenteSubtitulo; private String tamanioFuenteSubtitulo;
private String usuario;  private String colorFondoPagina;
private String colorBorde;  private String colorFondoFormulario;
private FormFile imagenFondoPagina;  private FormFile imagenCabecera;

private String tipoFuenteCombo;    private String colorFuenteCombo;    private String
colorFondoCombo;   private String tipoFuenteLista;   private String colorFuenteLista;
private String colorFondoLista; private String tipoFuenteBoton;  private String colorFuenteBoton; private
String colorFondoBoton;private String estadoEtiquetacolor="0"; private String
estadoTitulocolor="0";private String estadoSubtitulocolor ="0";private String estadoEtiquetafuente ="0";
private String estadoTitulofuente ="0"; private String estadoSubtitulofuente ="0";  private String
estadoEtiquetatamanio ="0"; private String estadoTitulotamanio="0"; private String
estadoSubtitulotamanio ="0";  private String estadoListacolor ="0";  private String estadoListafuente
="0";  private String estadoListatamanio ="0";  private String estadoCombocolor="0";  private String
estadoCombofuente ="0"; private String estadoCombotamanio ="0"; private String
estadoColorformulario ="0"; private String estadoBotonfuente ="0"; private String
estadoBotonColorFuente="0"; private String estadoBotontipofuente ="0"; private String
estadoColorPagina="0";  private String estadoColorBorde="0";  private String estadoImagenCabecera
="0";    public EstiloFormulario() {    }  public void setTipoFuenteEtiqueta(String tipoFuenteEtiqueta) {
this.tipoFuenteEtiqueta = tipoFuenteEtiqueta;   }
public String getTipoFuenteEtiqueta() {    return tipoFuenteEtiqueta;   }
public void setColorFuenteEtiqueta(String colorFuenteEtiqueta) {    this.colorFuenteEtiqueta =
colorFuenteEtiqueta;  }   public String getColorFuenteEtiqueta() {    return colorFuenteEtiqueta;   }
public void setTamanioFuenteEtiqueta(String tamanioFuenteEtiqueta) {    this.tamanioFuenteEtiqueta
= tamanioFuenteEtiqueta;   }   public String getTamanioFuenteEtiqueta() {    return
tamanioFuenteEtiqueta;  }
public void setTipoFuenteTitulo(String tipoFuenteTitulo) {    this.tipoFuenteTitulo = tipoFuenteTitulo;  }
public String getTipoFuenteTitulo() {    return tipoFuenteTitulo;  }
public void setColorFuenteTitulo(String colorFuenteTitulo) {    this.colorFuenteTitulo =
colorFuenteTitulo;   }  public String getColorFuenteTitulo() {
```

```
return colorFuenteTitulo; }public void setTamanioFuenteTitulo(String tamanioFuenteTitulo) {
this.tamanioFuenteTitulo = tamanioFuenteTitulo; }
public String getTamanioFuenteTitulo() { return tamanioFuenteTitulo; }
public void setTipoFuenteSubtitulo(String tipoFuenteSubtitulo) { this.tipoFuenteSubtitulo =
tipoFuenteSubtitulo; } public String getTipoFuenteSubtitulo() { return tipoFuenteSubtitulo; }
public void setColorFuenteSubtitulo(String colorFuenteSubtitulo) { this.colorFuenteSubtitulo =
colorFuenteSubtitulo; } public String getColorFuenteSubtitulo() { return colorFuenteSubtitulo; }
public void setTamanioFuenteSubtitulo(String tamanioFuenteSubtitulo) { this.tamanioFuenteSubtitulo
= tamanioFuenteSubtitulo;
} public String getTamanioFuenteSubtitulo() { return tamanioFuenteSubtitulo; }
public void setUsuario(String usuario) { this.usuario = usuario; } public String getUsuario() { return
usuario; } public void setColorFondoPagina(String colorFondoPagina) { this.colorFondoPagina =
colorFondoPagina; }
public String getColorFondoPagina() { return colorFondoPagina; }
public void setColorFondoFormulario(String colorFondoFormulario) { this.colorFondoFormulario =
colorFondoFormulario; } public String getColorFondoFormulario() { return colorFondoFormulario; }

   public void setColorBorde(String colorBorde) { this.colorBorde = colorBorde; }
   public String getColorBorde() { return colorBorde; }
  public void setTipoFuenteCombo(String tipoFuenteCombo) { this.tipoFuenteCombo =
tipoFuenteCombo; } public String getTipoFuenteCombo() { return tipoFuenteCombo; }
   public void setColorFuenteCombo(String colorFuenteCombo) { this.colorFuenteCombo =
colorFuenteCombo; }
   public String getColorFuenteCombo() { return colorFuenteCombo; }
   public void setColorFondoCombo(String colorFondoCombo) { thiscolorFondoCombo =
colorFondoCombo; }
   public String getColorFondoCombo() { return colorFondoCombo; }
   public void setCodigoPlantilla(int codigoPlantilla) { this.codigoPlantilla = codigoPlantilla; }
   public int getCodigoPlantilla() { return codigoPlantilla; }
   public void setTipoFuenteLista(String tipoFuenteLista) { this.tipoFuenteLista = tipoFuenteLista;
}
   public String getTipoFuenteLista() { return tipoFuenteLista; }
   public void setColorFuenteLista(String colorFuenteLista) { this.colorFuenteLista =
colorFuenteLista; }
   public String getColorFuenteLista() { return colorFuenteLista; }
   public void setColorFondoLista(String colorFondoLista) { this.colorFondoLista = colorFondoLista;
}
   public String getColorFondoLista() { return colorFondoLista; }
   public void setTipoFuenteBoton(String tipoFuenteBoton) { this.tipoFuenteBoton =
tipoFuenteBoton; }
   public String getTipoFuenteBoton() {
      return tipoFuenteBoton; }
   public void setColorFuenteBoton(String colorFuenteBoton) { this.colorFuenteBoton =
colorFuenteBoton; }
   public String getColorFuenteBoton() { return colorFuenteBoton; }
   public void setColorFondoBoton(String colorFondoBoton) { this.colorFondoBoton =
colorFondoBoton; }
   public String getColorFondoBoton() { return colorFondoBoton; }
   public FormFile getImagenFondoPagina() { return imagenFondoPagina;
   } public void setImagenFondoPagina(FormFile imagenFondoPagina) {
this.imagenFondoPagina = imagenFondoPagina; }
   public FormFile getImagenCabecera() { return imagenCabecera; }
   public void setImagenCabecera(FormFile imagenCabecera) { this.imagenCabecera =
imagenCabecera; }
   public void setEstadoEtiquetacolor(String estadoEtiquetacolor) { this.estadoEtiquetacolor =
estadoEtiquetacolor; }
   public String getEstadoEtiquetacolor() { return estadoEtiquetacolor;
   } public void setEstadoTitulocolor(String estadoTitulocolor) { this.estadoTitulocolor =
estadoTitulocolor; }
```

```
   public String getEstadoTitulocolor() {       return estadoTitulocolor;    }
   public void setEstadoSubtitulocolor(String estadoSubtitulocolor) {          this.estadoSubtitulocolor =
estadoSubtitulocolor;    }
   public String getEstadoSubtitulocolor() {       return estadoSubtitulocolor;    }

   public void setEstadoEtiquetafuente(String estadoEtiquetafuente) {
      this.estadoEtiquetafuente = estadoEtiquetafuente;    }
   public String getEstadoEtiquetafuente() {       return estadoEtiquetafuente;    }
   public void setEstadoTitulofuente(String estadoTitulofuente) {          this.estadoTitulofuente =
estadoTitulofuente;    }
   public String getEstadoTitulofuente() {       return estadoTitulofuente;    }
   public void setEstadoSubtitulofuente(String estadoSubtitulofuente) {       this.estadoSubtitulofuente =
estadoSubtitulofuente;    }
   public String getEstadoSubtitulofuente() {       return estadoSubtitulofuente;    }
   public void setEstadoEtiquetatamanio(String estadoEtiquetatamanio) {
this.estadoEtiquetatamanio = estadoEtiquetatamanio;    }
   public String getEstadoEtiquetatamanio() {       return estadoEtiquetatamanio;    }
   public void setEstadoTitulotamanio(String estadoTitulotamanio) {          this.estadoTitulotamanio =
estadoTitulotamanio;    }
   public String getEstadoTitulotamanio() {          return estadoTitulotamanio;      }      public void
setEstadoSubtitulotamanio(String estadoSubtitulotamanio) {
      this.estadoSubtitulotamanio = estadoSubtitulotamanio;    }
   public String getEstadoSubtitulotamanio() {       return estadoSubtitulotamanio;    }
   public void setEstadoListacolor(String estadoListacolor) {          this.estadoListacolor =
estadoListacolor;    }
   public String getEstadoListacolor() {       return estadoListacolor;    }
   public void setEstadoListafuente(String estadoListafuente) {          this.estadoListafuente =
estadoListafuente;    }
   public String getEstadoListafuente() {       return estadoListafuente;    }
   public void setEstadoListatamanio(String estadoListatamanio) {
      this.estadoListatamanio = estadoListatamanio;    }
   public String getEstadoListatamanio() {       return estadoListatamanio;    }
   public void setEstadoCombocolor(String estadoCombocolor) {          this.estadoCombocolor =
estadoCombocolor;    }
   public String getEstadoCombocolor() {       return estadoCombocolor;    }
   public void setEstadoCombofuente(String estadoCombofuente) {          this.estadoCombofuente =
estadoCombofuente;    }    public String getEstadoCombofuente() {       return estadoCombofuente;    }
   public void setEstadoCombotamanio(String estadoCombotamanio) {       this.estadoCombotamanio
= estadoCombotamanio;    }
   public String getEstadoCombotamanio() {       return estadoCombotamanio;    }
   public void setEstadoColorformulario(String estadoColorformulario) {       this.estadoColorformulario
= estadoColorformulario;    }
   public String getEstadoColorformulario() {       return estadoColorformulario;    }
   public void setEstadoBotonfuente(String estadoBotonfuente) {          this.estadoBotonfuente =
estadoBotonfuente;    }
   public String getEstadoBotonfuente() {          return estadoBotonfuente;      }      public void
setEstadoBotonColorFuente(String estadoBotonColorFuente) {
      this.estadoBotonColorFuente = estadoBotonColorFuente;    }
   public String getEstadoBotonColorFuente() {       return estadoBotonColorFuente;    }    public void
setEstadoBotontipofuente(String estadoBotontipofuente) {
      this.estadoBotontipofuente = estadoBotontipofuente;    }
   public String getEstadoBotontipofuente() {       return estadoBotontipofuente;    }    public void
setEstadoColorPagina(String estadoColorPagina) {
      this.estadoColorPagina = estadoColorPagina;    }
   public String getEstadoColorPagina() {       return estadoColorPagina;    }
   public void setEstadoColorBorde(String estadoColorBorde) {
      this.estadoColorBorde = estadoColorBorde;    }
   public String getEstadoColorBorde() {       return estadoColorBorde;
   }    public void setEstadoImagenCabecera(String estadoImagenCabecera) {
```

```
    this.estadoImagenCabecera = estadoImagenCabecera;    }    public String
getEstadoImagenCabecera() {    return estadoImagenCabecera;
  }}
```

BuscarObjetosServlets.java

Descripción: Clase por la cual se recupera los objetos de personalización para definir el estilo e invoca como respuesta el EstiloPropio.jsp.

Enlaces: ColorFuenteBO, TamanioFuenteBO, TipoFuenteBO.

```
package com.bvg.servlets;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.*;
import javax.servlet.http.*;
import com.bvg.bo.ColorFuenteBO;
import com.bvg.bo.TamanioFuenteBO;
import com.bvg.bo.TipoFuenteBO;
import java.util.Vector;

public class BuscarObjetosServlet extends HttpServlet {
  private static final String CONTENT_TYPE =
    "text/html; charset=windows-1252";
  public void init(ServletConfig config) throws ServletException {
    super.init(config);   }
  public void service(HttpServletRequest request,
          HttpServletResponse    response)    throws    ServletException,
IOException {
    response.setContentType(CONTENT_TYPE);
    PrintWriter out = response.getWriter();
    out.println("<html>"); out.println("<head><title>BuscarObjetosServlet</title></head>");
    out.println("<body>");
    out.println("<p>The servlet has received a POST or GET. This is the reply.</p>");
out.println("</body></html>");    out.close();    response.setContentType("text/html");
ServletContext sc;
    sc = this.getServletContext();    ColorFuenteBO unColorFuenteBO = new ColorFuenteBO();
    Vector listaColorFuente = unColorFuenteBO.buscarTodos();
    request.setAttribute("listaColorFuente", listaColorFuente);
    TipoFuenteBO unTipoFuenteBO = new TipoFuenteBO();
    Vector listaTipoFuente = unTipoFuenteBO.buscarTodos();
    request.setAttribute("listaTipoFuente", listaTipoFuente);
    TamanioFuenteBO unTamanioFuenteBO = new TamanioFuenteBO();
    Vector listaTamanioFuente = unTamanioFuenteBO.buscarTodos();
    request.setAttribute("listaTamnioFuente", listaTamanioFuente);
    RequestDispatcher rd = sc.getRequestDispatcher("/EstiloPropio.jsp");
    rd.forward(request, response);  }}
```

EstiloPropioServlet.java

Descripción: Clase por la cual se recupera el estilo y el usuario e invoca como respuesta el EstiloPropio.jsp.

Bean: Persona.

```
package com.bvg.servlets;

import com.bvg.bean.Persona;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.*;
import javax.servlet.http.*;
public class EstiloPropioServlet extends HttpServlet {

    public EstiloPropioServlet() {
    }
    private static final String CONTENT_TYPE =
        "text/css; charset=windows-1252";

    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }

    public void service(HttpServletRequest request,
                    HttpServletResponse response) throws ServletException,
                                                   IOException {
        response.setContentType(CONTENT_TYPE);
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head><title>VerObjetospara estilo</title></head>");
        out.println("<body>");
        out.println("<p>The servlet has received a POST or GET. This is the reply.</p>");
out.println("</body></html>");
        out.close();        response.setContentType("text/css");
  ServletContext sc;        sc = this.getServletContext();
   Persona unaPersona = new Persona();

  String usuario = request.getParameter("usuario");
request.setAttribute("usuario", usuario);
        RequestDispatcher rd = sc.getRequestDispatcher("/EstiloPropio.jsp");
    rd.forward(request, response);   }}
```

**Util**

Base.java

Descripción: Clase por la cual se conecta a la base de datos Academico hecha en Postgres.

```
package com.bvg.util;
import java.sql.Connection;
import java.sql.DriverManager;
public class Base {
    private static Connection unCon = null;
    public static Connection getCon() {
        if (unCon == null) {           try {
                Class.forName("org.postgresql.Driver");unCon                                    =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/academico",
"postgres", "postgres");           } catch (Throwable e) {      System.out.println(e);          }       }       return
unCon;   }}
```

EscribirImagen.java

```
package com.bvg.util;
import com.bvg.bean.Imagen;
import com.bvg.bo.ImagenBO;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.IOException;
import org.apache.struts.upload.FormFile;
public class EscribirImagen {
```

Variables de imagenes

```
InputStream stream1;    InputStream stream2;    OutputStream bos;    byte[] archivoFotoBase;    public
EscribirImagen() {   }
    public String escribir(String usuario, int codigoImagenFoto) throws Throwable {
    String rutafoto = null;       ImagenBO unaImagenBO = new ImagenBO();          archivoFotoBase =
unaImagenBO.recuperarFoto(codigoImagenFoto);
        try {        String path = com.bvg.util.Parametro.ruta;       File f = new File(path);
            if (!f.isDirectory()) {       boolean b = f.mkdir();       }
            bos = new FileOutputStream(path + "/" + usuario + "foto.jpg");
            bos.write(archivoFotoBase);        bos.close();
            rutafoto = path + "/" + usuario + "foto.jpg";
            System.out.println("Se escribio el archivo " + rutafoto);
            rutafoto = "Imagenes/"+usuario+"foto.jpg";      System.out.println("src= " + rutafoto);     } catch
(FileNotFoundException fnfe) {      System.out.println("Error no se pudo escribir el archivito1");       }
return rutafoto;  }}
```

Action

ActualizarPlantillaAction

Trabaja con : Estilo Propio.jsp

package com.bvg.action;

Librerías :

```
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.Action;
import org.apache.struts.upload.FormFile;
public class ActualizarPlantillaAction extends Action {
   public ActualizarPlantillaAction() {    }    /* Action que valida y guarda la plantilla creada por el usuario*/
public ActionForward perform(ActionMapping mapping, ActionForm form,
ttpServletRequest request, HttpServletResponse response) throws IOException,
ServletException {    EstiloFormulario estiloFormulario = (EstiloFormulario)form;
       int codigoNuevaEtiqueta;
       int codigoNuevoTitulo;
       int codigoNuevoSubtitulo;
       int codigoNuevoBoton;
       int codigoNuevoFondoFormulario;
       int codigoNuevoFondoPagina;
       int codigoNuevaLista;
       int codigoNuevoCombo;
       int codigoNuevaImagenFoto;
       String banderaNuevaEtiqueta = "0";
       String banderaNuevoTitulo = "0";
       String banderaNuevoSubtitulo = "0";
       String banderaNuevoBoton = "0";
       String banderaNuevoFondoPagina = "0";
       String banderaNuevoFondoFormulario = "0";
       String banderaNuevoCombo = "0";
       String banderaNuevaLista = "0";
       boolean valorEtiqueta = false;
       boolean valorTitulo = false;
       boolean valorSubtitulo = false;
       boolean valorBoton = false;
       boolean valorFondoPagina = false;
       boolean valorFondoFormulario = false;
       boolean valorCombo = false;
       boolean valorLista = false;
       //variables de imagenes
       //InputStream stream;
       //OutputStream bos;
```

```
/*recibir datos de formualrio para validar --solo el usuario,
* los demás componentes del been se reciben directamente en los metodos*/
int codigoPlantilla = estiloFormulario.getCodigoPlantilla();
//CREAR LOS OBJETOS BO NUEVOS ETIQUETA, TITULO, SUBTITULO Y LA PLANTILLA
PlantillaBO unaPlantillaBO = new PlantillaBO();
EtiquetaBO unaEtiquetaBO = new EtiquetaBO();
TituloBO unTituloBO = new TituloBO();
SubtituloBO unSubtituloBO = new SubtituloBO();
//CREAR LOS OBJETOS BO NUEVO BOTON
BotonBO unBotonBO = new BotonBO();
//CREAR LOS OBJETOS BO NUEVOS FONDO DE PAGINA, FORNDO FORMULARIO, BORDE
E IMAGEN
FondoFormularioBO unFondoFormularioBO = new FondoFormularioBO();
FondoPaginaBO unFondoPaginaBO = new FondoPaginaBO();
//CREAR LOS OBJETOS BO NUEVOS COMBOS Y LISTAS
ComboBO unComboBO = new ComboBO();
ListasBO unaListaBO = new ListasBO();
/*Crear objeto BO para imagen*/
ImagenBO unaImagenBO = new ImagenBO();
/*utilizar los metodos de la clase BO para retornar*/
/*buscar la etiqueta*/
System.out.println("llega a Actualizar Plantilla");
int codigoEtiquetaR =
   unaEtiquetaBO.compararDatosEtiqueta(estiloFormulario.getTipoFuenteEtiqueta(),
                    estiloFormulario.getColorFuenteEtiqueta(),
                    estiloFormulario.getTamanioFuenteEtiqueta());
if (codigoEtiquetaR <= 0) {
   int codigoEtiqueta = unaEtiquetaBO.buscarCodigo();
   codigoEtiqueta = codigoEtiqueta + 1;      codigoNuevaEtiqueta = codigoEtiqueta;
   banderaNuevaEtiqueta = "1";    } else {
   codigoNuevaEtiqueta = codigoEtiquetaR;    }
/*buscar el titulo*/
int codigoTituloR =
   unTituloBO.compararDatosTitulo(estiloFormulario.getTipoFuenteTitulo(),
estiloFormulario.getColorFuenteTitulo(),
estiloFormulario.getTamanioFuenteTitulo());
if (codigoTituloR <= 0) {    int codigoTitulo = unTituloBO.buscarCodigo();
   codigoTitulo = codigoTitulo + 1;     codigoNuevoTitulo = codigoTitulo;
   banderaNuevoTitulo = "1";    } else {      codigoNuevoTitulo = codigoTituloR;    }
/*buscar subtitulo*/
int codigoSubtituloR =
   unSubtituloBO.compararDatosTitulo(estiloFormulario.getTipoFuenteSubtitulo(),
estiloFormulario.getColorFuenteSubtitulo(),       estiloFormulario.getTamanioFuenteSubtitulo());
if (codigoSubtituloR <= 0) {     int codigoSubtitulo = unSubtituloBO.buscarCodigo() + 1;
  codigoNuevoSubtitulo = codigoSubtitulo;       banderaNuevoSubtitulo = "1";    } else {
   codigoNuevoSubtitulo = codigoSubtituloR;    }
/*buscar boton*/
int codigoBotonR =
   unBotonBO.compararDatosBoton(estiloFormulario.getColorFondoBoton(),
estiloFormulario.getColorFuenteBoton(),       estiloFormulario.getTipoFuenteBoton());
if (codigoBotonR <= 0) {    int codigoBotonNuevo = unBotonBO.buscarCodigo() + 1;
   codigoNuevoBoton = codigoBotonNuevo;     banderaNuevoBoton = "1";    } else {
   codigoNuevoBoton = codigoBotonR;    }
/*buscar fondo de pagina*/
int codigoFondoPaginaR =
   unFondoPaginaBO.compararDatosFondoPagina(estiloFormulario.getColorFondoPagina());
if (codigoFondoPaginaR <= 0) {
   int codigoFondoPaginaNuevo = unFondoPaginaBO.buscarCodigo() + 1;
   codigoNuevoFondoPagina = codigoFondoPaginaNuevo;  banderaNuevoFondoPagina = "1";
```

```
} else {        codigoNuevoFondoPagina = codigoFondoPaginaR;      }
/*buscar fondo de formulario*/
int codigoFondoFormularioR =
unFondoFormularioBO.compararDatosFondoFormulario(estiloFormulario.getColorFondoFormulario());
if (codigoFondoFormularioR <= 0) {          int codigoFondoFormularioNuevo =
        unFondoFormularioBO.buscarCodigo() + 1;
    codigoNuevoFondoFormulario = codigoFondoFormularioNuevo;
    banderaNuevoFondoFormulario = "1";        } else {
    codigoNuevoFondoFormulario = codigoFondoFormularioR;        }
/*buscar combo*/        int codigoComboR =
    unComboBO.compararDatosCombo(estiloFormulario.getColorFondoCombo(),
estiloFormulario.getColorFuenteCombo(),               estiloFormulario.getTipoFuenteCombo());
if (codigoComboR <= 0) {           int codigoComboNuevo = unComboBO.buscarCodigo() + 1;
    codigoNuevoCombo = codigoComboNuevo;           banderaNuevoCombo = "1";
} else {           codigoNuevoCombo = codigoComboR;        }

/*buscar listas*/     int codigoListasR =
    unaListaBO.compararDatosLista(estiloFormulario.getColorFondoLista(),
estiloFormulario.getColorFuenteLista(),            estiloFormulario.getTipoFuenteLista());
if (codigoListasR <= 0) {    int codigoListaNuevo = unaListaBO.buscarCodigo() + 1;
    codigoNuevaLista = codigoListaNuevo;    banderaNuevaLista = "1";    } else {
    codigoNuevaLista = codigoListasR;     }
/*subir imagen*/      FormFile myFile1;      myFile1 = estiloFormulario.getImagenCabecera();
    /*Buscar ultimo codigo de imagen guardado*/
int codigoImagenR=unaImagenBO.recuperarCodigoImagenFotoPlantilla(codigoPlantilla);
if(codigoImagenR<=0){           codigoNuevaImagenFoto = unaImagenBO.buscarCodigo() + 1;
}else{           codigoNuevaImagenFoto = codigoImagenR;       }
/*Crear objeto imagen*/        if (myFile1.getFileName()!=""){
System.out.println(myFile1.getFileName());      Imagen unaImagenFoto = new Imagen();
unaImagenFoto.setCodigoImagen(codigoNuevaImagenFoto);
unaImagenFoto.setDescripción(com.bvg.util.Parametro.ruta + "/" +
estiloFormulario.getUsuario() +   "foto.jpg");    byte[] buffer;        buffer = myFile1.getFileData();
unaImagenFoto.setArchivo(buffer);       //guarda la foto en la BD         if(codigoImagenR<=0){
        unaImagenBO.guardar(unaImagenFoto);    }else{
        unaImagenBO.actualizar(unaImagenFoto);         }
    }          /*Crear el objeto nuevo de etiqueta que recibe el tipo, color, tamaniio y estilo de
fuente*/        Etiqueta unaEtiquetaNueva = new Etiqueta();
unaEtiquetaNueva.setCodigoEtiqueta(codigoNuevaEtiqueta);
unaEtiquetaNueva.setCodigoTipoFuente(estiloFormulario.getTipoFuenteEtiqueta());
unaEtiquetaNueva.setCodigoColorFuente(estiloFormulario.getColorFuenteEtiqueta());
unaEtiquetaNueva.setCodigoTamanioFuente(estiloFormulario.getTamanioFuenteEtiqueta());
/*Crear el objeto nuevo de titulo*/      Titulo unTituloNuevo = new Titulo();
unTituloNuevo.setCodigoTitulo(codigoNuevoTitulo);
unTituloNuevo.setCodigoTipoFuente(estiloFormulario.getTipoFuenteTitulo());
unTituloNuevo.setCodigoColorFuente(estiloFormulario.getColorFuenteTitulo());
unTituloNuevo.setCodigoTamanioFuente(estiloFormulario.getTamanioFuenteTitulo());
/*Crear el objeto nuevo de subtitulo*/
Subtitulo unSubtituloNuevo = new Subtitulo();
unSubtituloNuevo.setCodigoSubtitulo(codigoNuevoSubtitulo);
unSubtituloNuevo.setCodigoTipoFuente(estiloFormulario.getTipoFuenteSubtitulo());
unSubtituloNuevo.setCodigoColorFuente(estiloFormulario.getColorFuenteSubtitulo());
unSubtituloNuevo.setCodigoTamanioFuente(estiloFormulario.getTamanioFuenteSubtitulo());
/*Crear el objeto nuevo Boton*/
Boton unBotonNuevo = new Boton();
unBotonNuevo.setCodigoBoton(codigoNuevoBoton);
unBotonNuevo.setCodigoTipoFuente(estiloFormulario.getTipoFuenteBoton());
unBotonNuevo.setCodigoColorFondo(estiloFormulario.getColorFondoBoton());
unBotonNuevo.setCodigoColorFuente(estiloFormulario.getColorFuenteBoton());
/*Crear el objeto nuevo de FondoPagina*/
```

```
FondoPagina unFondoPaginaNuevo = new FondoPagina();
unFondoPaginaNuevo.setCodigoFondoPagina(codigoNuevoFondoPagina);
unFondoPaginaNuevo.setCodigoColorFondo(estiloFormulario.getColorFondoPagina());

/*Crear el objeto nuevo de FondoFormulario*/
FondoFormulario unFondoFormularioNuevo = new FondoFormulario();
unFondoFormularioNuevo.setCodigoFondoFormulario(codigoNuevoFondoFormulario);
unFondoFormularioNuevo.setCodigoColorFondo(estiloFormulario.getColorFondoFormulario());

/*Crear el objeto nuevo Listas*/
Listas unaListaNueva = new Listas();
unaListaNueva.setCodigocolorfondo(estiloFormulario.getColorFondoLista());
unaListaNueva.setCodigocolorfuente(estiloFormulario.getColorFuenteLista());
unaListaNueva.setCodigoTipoFuente(estiloFormulario.getTipoFuenteLista());
unaListaNueva.setCodigoLista(codigoNuevaLista);

/*Crear el objeto nuevo Combo*/
Combo unComboNuevo = new Combo();
unComboNuevo.setCodigoCombo(codigoNuevoCombo);
unComboNuevo.setCodigotipofuente(estiloFormulario.getTipoFuenteCombo());
unComboNuevo.setCodigocolorfondo(estiloFormulario.getColorFondoCombo());
unComboNuevo.setCodigocolorfuente(estiloFormulario.getColorFuenteCombo());

/*Crear el objeto nuevo de PlantillaCodigo de la base de datos*/
PlantillaCodigo unaPlantillaCodigo = new PlantillaCodigo();
int codigoNuevaPlantilla = codigoPlantilla;
unaPlantillaCodigo.setCodigoPlantilla(codigoNuevaPlantilla);
unaPlantillaCodigo.setCodigoEtiqueta(codigoNuevaEtiqueta);
unaPlantillaCodigo.setCodigoTitulo(codigoNuevoTitulo);
unaPlantillaCodigo.setCodigoSubtitulo(codigoNuevoSubtitulo);
unaPlantillaCodigo.setCodigoBoton(codigoNuevoBoton);
unaPlantillaCodigo.setCodigoFondoPagina(codigoNuevoFondoPagina);
unaPlantillaCodigo.setCodigoFondoFormulario(codigoNuevoFondoFormulario);
unaPlantillaCodigo.setCodigoCombo(codigoNuevoCombo);
unaPlantillaCodigo.setCodigoLista(codigoNuevaLista);
unaPlantillaCodigo.setCodigoImagenFoto(codigoNuevaImagenFoto);

/*Guardar o Actualizar los objetos creados*/
if (banderaNuevaEtiqueta == "1") {
   valorEtiqueta = unaEtiquetaBO.guardar(unaEtiquetaNueva);       }
if (banderaNuevoTitulo == "1") {
   valorTitulo = unTituloBO.guardar(unTituloNuevo);       }
if (banderaNuevoSubtitulo == "1") {
   valorSubtitulo = unSubtituloBO.guardar(unSubtituloNuevo);       }
if (banderaNuevoBoton == "1") {
   valorBoton = unBotonBO.guardar(unBotonNuevo);       }
if (banderaNuevoFondoPagina == "1") {
   valorFondoPagina = unFondoPaginaBO.guardar(unFondoPaginaNuevo);       }
if (banderaNuevoFondoFormulario == "1") {
   valorFondoFormulario =
         unFondoFormularioBO.guardar(unFondoFormularioNuevo);       }
if (banderaNuevoCombo == "1") {
   valorCombo = unComboBO.guardar(unComboNuevo);       }
if (banderaNuevaLista == "1") {
   valorLista = unaListaBO.guardar(unaListaNueva);       }
boolean valorPlantilla = unaPlantillaBO.actualizar(unaPlantillaCodigo);
if (valorPlantilla == false) {
   System.out.println("Error al guardar objetos");       }
return mapping.findForward("success");    }}
```

CrearPlantillaAction

package com.bvg.action;

Trabaja con: EstiloPropio.jsp

Librerías :

```
import com.bvg.bean.Boton;
import com.bvg.bean.Combo;
import com.bvg.bean.Etiqueta;
import com.bvg.bean.FondoFormulario;
import com.bvg.bean.FondoPagina;
import com.bvg.bean.Imagen;
import com.bvg.bean.Listas;
import com.bvg.bean.PlantillaCodigo;
import com.bvg.bean.Subtitulo;
import com.bvg.bean.Titulo;
import com.bvg.bo.BotonBO;
import com.bvg.bo.ComboBO;
import com.bvg.bo.EtiquetaBO;
import com.bvg.bo.FondoFormularioBO;
import com.bvg.bo.FondoPaginaBO;
import com.bvg.bo.ImagenBO;
import com.bvg.bo.ListasBO;
import com.bvg.bo.PersonaBO;
import com.bvg.bo.PlantillaBO;
import com.bvg.bo.TituloBO;
import com.bvg.bo.SubtituloBO;
import com.bvg.formulario.EstiloFormulario;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.Action;
import org.apache.struts.upload.FormFile;
public class CrearPlantillaAction extends Action {
    public CrearPlantillaAction() {    }
    /* Action que valida y guarda la plantilla creada por el usuario*/
    public ActionForward perform(ActionMapping mapping, ActionForm form,
   HttpServletRequest request,           HttpServletResponse response) throws IOException,
 ServletException {      EstiloFormulario estiloFormulario = (EstiloFormulario)form;
        int codigoNuevaEtiqueta;
        int codigoNuevoTitulo;
        int codigoNuevoSubtitulo;
        int codigoNuevoBoton;
        int codigoNuevoFondoFormulario;
        int codigoNuevoFondoPagina;
        int codigoNuevaLista;
        int codigoNuevoCombo;
        int codigoNuevaImagenFoto;
        String banderaNuevaEtiqueta = "0";
        String banderaNuevoTitulo = "0";
        String banderaNuevoSubtitulo = "0";
        String banderaNuevoBoton = "0";
```

```
        String banderaNuevoFondoPagina = "0";
        String banderaNuevoFondoFormulario = "0";
        String banderaNuevoCombo = "0";
        String banderaNuevaLista = "0";
        boolean valorEtiqueta = false;
        boolean valorTitulo = false;
        boolean valorSubtitulo = false;
        boolean valorBoton = false;
        boolean valorFondoPagina = false;
        boolean valorFondoFormulario = false;
        boolean valorCombo = false;
        boolean valorLista = false;
        /*recibir datos de formualrio para validar --solo el usuario,
      * los demás componentes del been se reciben directamente en los metodos*/
        String usuario = estiloFormulario.getUsuario();
        int codigoPlantilla = estiloFormulario.getCodigoPlantilla();

        //CREAR LOS OBJETOS BO NUEVOS ETIQUETA, TITULO Y SUBTITULO
        PlantillaBO unaPlantillaBO = new PlantillaBO();
        EtiquetaBO unaEtiquetaBO = new EtiquetaBO();
        TituloBO unTituloBO = new TituloBO();
        SubtituloBO unSubtituloBO = new SubtituloBO();
        PersonaBO unaPersonaBO = new PersonaBO();
        //CREAR OBJETO BO NUEVO PARA BOTON
        BotonBO unBotonBO = new BotonBO();
//CREAR LOS OBJETOS BO NUEVOS FONDO DE PAGINA, FORNDO FORMULARIO, BORDE E IMAGEN
        FondoFormularioBO unFondoFormularioBO = new FondoFormularioBO();
        FondoPaginaBO unFondoPaginaBO = new FondoPaginaBO();
        //CREAR LOS OBJETOS BO NUEVOS COMBOS Y LISTAS
        ComboBO unComboBO = new ComboBO();
        ListasBO unaListaBO = new ListasBO();
        /*Crear objeto BO para imagen*/
        ImagenBO unaImagenBO = new ImagenBO();
        /*utilizar los metodos de la clase BO para retornar*/
        System.out.println("Llega a CrearPlantillaAction");
        /*buscar la etiqueta*/
        int codigoEtiquetaR =
            unaEtiquetaBO.compararDatosEtiqueta(estiloFormulario.getTipoFuenteEtiqueta(),
                                estiloFormulario.getColorFuenteEtiqueta(),
                                estiloFormulario.getTamanioFuenteEtiqueta());
        if (codigoEtiquetaR <= 0) {
            int codigoEtiqueta = unaEtiquetaBO.buscarCodigo();
            codigoEtiqueta = codigoEtiqueta + 1;
            codigoNuevaEtiqueta = codigoEtiqueta;   banderaNuevaEtiqueta = "1";        } else {
codigoNuevaEtiqueta = codigoEtiquetaR;      }       /*buscar el titulo*/
        int codigoTituloR =
            unTituloBO.compararDatosTitulo(estiloFormulario.getTipoFuenteTitulo(),
estiloFormulario.getColorFuenteTitulo(),   estiloFormulario.getTamanioFuenteTitulo());
        if (codigoTituloR <= 0) {     int codigoTitulo = unTituloBO.buscarCodigo();
            codigoTitulo = codigoTitulo + 1; codigoNuevoTitulo = codigoTitulo;  banderaNuevoTitulo = "1";
        } else {    codigoNuevoTitulo = codigoTituloR;    }
        /*buscar subtitulo*/
        int codigoSubtituloR =
            unSubtituloBO.compararDatosTitulo(estiloFormulario.getTipoFuenteSubtitulo(),
                                estiloFormulario.getColorFuenteSubtitulo(),
                                estiloFormulario.getTamanioFuenteSubtitulo());
        if (codigoSubtituloR <= 0) {   int codigoSubtitulo = unSubtituloBO.buscarCodigo() + 1;
            codigoNuevoSubtitulo = codigoSubtitulo;     banderaNuevoSubtitulo = "1";    } else {
```

```
        codigoNuevoSubtitulo = codigoSubtituloR;        }
    /*buscar boton*/
    int codigoBotonR =
        unBotonBO.compararDatosBoton(estiloFormulario.getColorFondoBoton(),
                              estiloFormulario.getColorFuenteBoton(),
                              estiloFormulario.getTipoFuenteBoton());
    if (codigoBotonR <= 0) {        int codigoBotonNuevo = unBotonBO.buscarCodigo() + 1;
        codigoNuevoBoton = codigoBotonNuevo;        banderaNuevoBoton = "1";
    } else {        codigoNuevoBoton = codigoBotonR;        }
    /*buscar fondo de pagina*/
    int codigoFondoPaginaR =
        unFondoPaginaBO.compararDatosFondoPagina(estiloFormulario.getColorFondoPagina());
    if (codigoFondoPaginaR <= 0) {
        int codigoFondoPaginaNuevo = unFondoPaginaBO.buscarCodigo() + 1;
        codigoNuevoFondoPagina = codigoFondoPaginaNuevo;
        banderaNuevoFondoPagina = "1";        } else {
        codigoNuevoFondoPagina = codigoFondoPaginaR;
    }
    /*buscar fondo de formulario*/
    int codigoFondoFormularioR =
unfondoFormularioBO.compararDatosFondoFormulario(estiloFormulario.getColorFondoFormulario());
    if (codigoFondoFormularioR <= 0) {
        int codigoFondoFormularioNuevo =
            unFondoFormularioBO.buscarCodigo() + 1;
        codigoNuevoFondoFormulario = codigoFondoFormularioNuevo;
        banderaNuevoFondoFormulario = "1";        } else {        codigoNuevoFondoFormulario =
codigoFondoFormularioR;        }
    /*buscar combo*/
    int codigoComboR =
        unComboBO.compararDatosCombo(estiloFormulario.getColorFondoCombo(),
                              estiloFormulario.getColorFuenteCombo(),
                              estiloFormulario.getTipoFuenteCombo());
    if (codigoComboR <= 0) {        int codigoComboNuevo = unComboBO.buscarCodigo() + 1;
        codigoNuevoCombo = codigoComboNuevo;        banderaNuevoCombo = "1";
    } else {        codigoNuevoCombo = codigoComboR;        }
    /*buscar listas*/
    int codigoListasR =
        unaListaBO.compararDatosLista(estiloFormulario.getColorFondoLista(),
                              estiloFormulario.getColorFuenteLista(),
                              estiloFormulario.getTipoFuenteLista());
    if (codigoListasR <= 0) {        int codigoListaNuevo = unaListaBO.buscarCodigo() + 1;
        codigoNuevaLista = codigoListaNuevo;        banderaNuevaLista = "1";
    } else {        codigoNuevaLista = codigoListasR;
    }        /*subir imagen*/
    FormFile myFile1;
    myFile1 = estiloFormulario.getImagenCabecera();
    System.out.println(myFile1.getFileName());
    /*Buscar ultimo codigo de imagen guardado*/
    codigoNuevaImagenFoto = unaImagenBO.buscarCodigo() + 1;
    /*Crear objeto imagen*/
    if (myFile1!=null){
    Imagen unaImagenFoto = new Imagen();
    unaImagenFoto.setCodigoImagen(codigoNuevaImagenFoto);
    unaImagenFoto.setDescripción(com.bvg.util.Parametro.ruta + "/" +
                          estiloFormulario.getUsuario() +
                          "foto.jpg");
    byte[] buffer;
    buffer = myFile1.getFileData();
    unaImagenFoto.setArchivo(buffer);
```

```
//guarda la foto en la BD
unaImagenBO.guardar(unaImagenFoto);
}
/*Crear el objeto nuevo de etiqueta que recibe el tipo, color, tamaniio y estilo de fuente*/
Etiqueta unaEtiquetaNueva = new Etiqueta();
unaEtiquetaNueva.setCodigoEtiqueta(codigoNuevaEtiqueta);
unaEtiquetaNueva.setCodigoTipoFuente(estiloFormulario.getTipoFuenteEtiqueta());
unaEtiquetaNueva.setCodigoColorFuente(estiloFormulario.getColorFuenteEtiqueta());
unaEtiquetaNueva.setCodigoTamanioFuente(estiloFormulario.getTamanioFuenteEtiqueta());
/*Crear el objeto nuevo de titulo*/
Titulo unTituloNuevo = new Titulo();
unTituloNuevo.setCodigoTitulo(codigoNuevoTitulo);
unTituloNuevo.setCodigoTipoFuente(estiloFormulario.getTipoFuenteTitulo());
unTituloNuevo.setCodigoColorFuente(estiloFormulario.getColorFuenteTitulo());
unTituloNuevo.setCodigoTamanioFuente(estiloFormulario.getTamanioFuenteTitulo());


/*Crear el objeto nuevo de subtitulo*/
Subtitulo unSubtituloNuevo = new Subtitulo();
unSubtituloNuevo.setCodigoSubtitulo(codigoNuevoSubtitulo);
unSubtituloNuevo.setCodigoTipoFuente(estiloFormulario.getTipoFuenteSubtitulo());
unSubtituloNuevo.setCodigoColorFuente(estiloFormulario.getColorFuenteSubtitulo());
unSubtituloNuevo.setCodigoTamanioFuente(estiloFormulario.getTamanioFuenteSubtitulo());
/*Crear el objeto nuevo Boton*/
Boton unBotonNuevo = new Boton();
unBotonNuevo.setCodigoBoton(codigoNuevoBoton);
unBotonNuevo.setCodigoTipoFuente(estiloFormulario.getTipoFuenteBoton());
unBotonNuevo.setCodigoColorFondo(estiloFormulario.getColorFondoBoton());
unBotonNuevo.setCodigoColorFuente(estiloFormulario.getColorFuenteBoton());
/*Crear el objeto nuevo de FondoPagina*/
FondoPagina unFondoPaginaNuevo = new FondoPagina();
unFondoPaginaNuevo.setCodigoFondoPagina(codigoNuevoFondoPagina);
unFondoPaginaNuevo.setCodigoColorFondo(estiloFormulario.getColorFondoPagina());
//unFondoPaginaNuevo.setCodigoImagen(estiloFormulario.getImagenFondoPagina());
/*Crear el objeto nuevo de FondoFormulario*/
FondoFormulario unFondoFormularioNuevo = new FondoFormulario();
unFondoFormularioNuevo.setCodigoFondoFormulario(codigoNuevoFondoFormulario);
unFondoFormularioNuevo.setCodigoColorFondo(estiloFormulario.getColorFondoFormulario());
/*Crear el objeto nuevo Listas*/
Listas unaListaNueva = new Listas();
unaListaNueva.setCodigocolorfondo(estiloFormulario.getColorFondoLista());
unaListaNueva.setCodigocolorfuente(estiloFormulario.getColorFuenteLista());
unaListaNueva.setCodigoTipoFuente(estiloFormulario.getTipoFuenteLista());
unaListaNueva.setCodigoLista(codigoNuevaLista);
/*Crear el objeto nuevo Combo*/
Combo unComboNuevo = new Combo();
unComboNuevo.setCodigoCombo(codigoNuevoCombo);
unComboNuevo.setCodigotipofuente(estiloFormulario.getTipoFuenteCombo());
unComboNuevo.setCodigocolorfondo(estiloFormulario.getColorFondoCombo());
unComboNuevo.setCodigocolorfuente(estiloFormulario.getColorFuenteCombo());
/*Crear el objeto nuevo de PlantillaCodigo de la base de datos*/
PlantillaCodigo unaPlantillaCodigo = new PlantillaCodigo();
int codigoNuevaPlantilla = codigoPlantilla;
unaPlantillaCodigo.setCodigoPlantilla(codigoNuevaPlantilla);
unaPlantillaCodigo.setCodigoEtiqueta(codigoNuevaEtiqueta);
unaPlantillaCodigo.setCodigoTitulo(codigoNuevoTitulo);
unaPlantillaCodigo.setCodigoSubtitulo(codigoNuevoSubtitulo);
unaPlantillaCodigo.setCodigoBoton(codigoNuevoBoton);
unaPlantillaCodigo.setCodigoFondoPagina(codigoNuevoFondoPagina);
```

```
        unaPlantillaCodigo.setCodigoFondoFormulario(codigoNuevoFondoFormulario);
        unaPlantillaCodigo.setCodigoCombo(codigoNuevoCombo);
        unaPlantillaCodigo.setCodigoLista(codigoNuevaLista);
        /*Guardar o Actualizar los objetos creados*/
        if (banderaNuevaEtiqueta == "1") {
           valorEtiqueta = unaEtiquetaBO.guardar(unaEtiquetaNueva);     }
        if (banderaNuevoTitulo == "1") {
           valorTitulo = unTituloBO.guardar(unTituloNuevo);     }
        if (banderaNuevoSubtitulo == "1") {
           valorSubtitulo = unSubtituloBO.guardar(unSubtituloNuevo);     }
        if (banderaNuevoFondoPagina == "1") {
           valorFondoPagina = unFondoPaginaBO.guardar(unFondoPaginaNuevo);     }
        if (banderaNuevoBoton == "1") {
           valorBoton = unBotonBO.guardar(unBotonNuevo);     }
        if (banderaNuevoFondoFormulario == "1") {
           valorFondoFormulario =
                 unFondoFormularioBO.guardar(unFondoFormularioNuevo);
        }     if (banderaNuevoCombo == "1") {
           valorCombo = unComboBO.guardar(unComboNuevo);     }
        if (banderaNuevaLista == "1") {
           valorLista = unaListaBO.guardar(unaListaNueva);     }
        boolean valorPlantilla = unaPlantillaBO.guardar(unaPlantillaCodigo);
        boolean valorUsuarioPlantilla =
           unaPersonaBO.guardarPlantilla(usuario, codigoNuevaPlantilla);
        if (valorPlantilla == false) {     System.out.println("No se guardo nuevo objeto Plantilla");     }
        if (valorEtiqueta == false) {     System.out.println("No se guardo nuevo objeto Etiqueta");     }
        if (valorTitulo == false) {     System.out.println("No se guardo nuevo objeto Titulo");     }
        if (valorSubtitulo == false) {     System.out.println("No se guardo nuevo objeto Subtitulo");     }
        if (valorBoton == false) {     System.out.println("No se guardo nuevo objeto Boton");     }
        if (valorLista == false) {     System.out.println("No se guardo nuevo objeto lista");     }
        if (valorCombo == false) {     System.out.println("No se guardo nuevo objeto Combo");     }
        if (valorUsuarioPlantilla == false) {     System.out.println("No se guardo nuevo en la tabla
usaurio el codigo de plantilla");     }
        if (valorFondoPagina == false) {     System.out.println("No se guardo nuevo objeto Fondo de
Pagina");     }     return mapping.findForward("success");   }}
```

EstiloPropioAction

Trabaja con: EstiloPropio.jsp

Librerías :

```
package com.bvg.action;
import com.bvg.bean.Persona;
import com.bvg.bo.PlantillaBO;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.Action;
public class EstiloPropioAction extends Action {
   public EstiloPropioAction() {   }
   /* Action que valida y guarda la plantilla creada por el usuario*/
   public ActionForward perform(ActionMapping mapping, ActionForm form,
```

```
                  HttpServletRequest request,
              HttpServletResponse response) throws IOException,
                                               ServletException {
    System.out.println("Entro a EstiloPropioAction");       String usuario;
    String banderaCrear;      int codigoPlantilla;
    HttpSession session = request.getSession(true);
    if (session != null) {         Persona unaPersona = (Persona)session.getAttribute("unaPersona");
        usuario = unaPersona.getUsuario();           PlantillaBO unaPlantillaBO = new PlantillaBO();
        codigoPlantilla = unaPlantillaBO.buscarCodigoPlantilla(usuario);
        System.out.println("codigoPlantilla" + codigoPlantilla);
        if (codigoPlantilla == 0) {
            //recupero el ultimo codigoplantilla creado y lo incremento
            codigoPlantilla = unaPlantillaBO.buscarCodigo() + 1;
            System.out.println("Crear nueva Plantilla");
            System.out.println("codigoNuevoPlantilla " + codigoPlantilla);
            banderaCrear = "1";
        } else {
            System.out.println("Actualiza Plantilla");
            banderaCrear = "0";           }
        unaPersona.setCodigoPlantilla(codigoPlantilla);
        session.setAttribute("unaPersonaP", unaPersona);
        session.setAttribute("banderaCrear", banderaCrear);       }
    return mapping.findForward("success");       //el succes es CrearPlantilla   }}
```

LoadPlantillaAction

Trabaja con: EstiloPropio.jsp

Librerías :

```
package com.bvg.action;

import com.bvg.bean.Persona;
import com.bvg.bo.PersonaBO;
import com.bvg.bean.Plantilla;
import com.bvg.bean.PlantillaCodigo;
import com.bvg.bo.PlantillaBO;
import com.bvg.form.PlantillaForma;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionError;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
public class LoadPlantillaAction extends Action {
  public LoadPlantillaAction() {    }  public ActionForward perform(ActionMapping mapping, ActionForm
form, HttpServletRequest request,                                  HttpServletResponse response) throws
IOException,                                        ServletException {   String usuario = null;
    Persona unaPersona = new Persona();
    ActionErrors errors = new ActionErrors();
    PlantillaForma unaPlantillaForm = (PlantillaForma)form;
    HttpSession session = request.getSession(true);
```

```
        if (session != null) {
            unaPersona = (Persona)session.getAttribute("unaPersona");
            usuario = unaPersona.getUsuario();
        }
        int codigoPlantilla = unaPlantillaForm.getCodigoPlantilla();
        if (errors.size() > 0) {
            saveErrors(request, errors);
            return new ActionForward(mapping.getInput());
        } else {
            PlantillaBO unaPlantillaBO = new PlantillaBO();
            PlantillaCodigo unaPlantillaCodigo =
                unaPlantillaBO.recuperarPlantilla(codigoPlantilla);
            PlantillaCodigo unaPlantillaCodigoNueva = new PlantillaCodigo();

            //recupero el ultimo codigoplantilla creado y lo incremento
            codigoPlantilla = unaPlantillaBO.buscarCodigo() + 1;
            unaPlantillaCodigoNueva.setCodigoPlantilla(codigoPlantilla);
            unaPlantillaCodigoNueva.setCodigoBoton(unaPlantillaCodigo.getCodigoBoton());
            unaPlantillaCodigoNueva.setCodigoCombo(unaPlantillaCodigo.getCodigoCombo());
            unaPlantillaCodigoNueva.setCodigoLista(unaPlantillaCodigo.getCodigoLista());
            unaPlantillaCodigoNueva.setCodigoEtiqueta(unaPlantillaCodigo.getCodigoEtiqueta());
            unaPlantillaCodigoNueva.setCodigoTitulo(unaPlantillaCodigo.getCodigoTitulo());
            unaPlantillaCodigoNueva.setCodigoSubtitulo(unaPlantillaCodigo.getCodigoSubtitulo());

unaPlantillaCodigoNueva.setCodigoFondoPagina(unaPlantillaCodigo.getCodigoFondoPagina());
unaPlantillaCodigoNueva.setCodigoFondoFormulario(unaPlantillaCodigo.getCodigoFondoFormulario())
 unaPlantillaCodigoNueva.setCodigoImagenHeader(unaPlantillaCodigo.getCodigoImagenHeader());
    unaPlantillaCodigoNueva.setCodigoImagenFooter(unaPlantillaCodigo.getCodigoImagenFooter());
            unaPlantillaCodigoNueva.setCodigoImagenBody(unaPlantillaCodigo.getCodigoImagenBody());
            boolean devolverP =
                unaPlantillaBO.guardar(unaPlantillaCodigoNueva);          if (devolverP == false) {
                System.out.println("Error al guardar la plantilla en la tabla plantilla en LoadPlantillaAction");  }
            System.out.println("codigoNuevoPlantilla " + codigoPlantilla);
            PersonaBO unaPersonaBO = new PersonaBO();
            boolean devolver =
                unaPersonaBO.guardarPlantilla(usuario, codigoPlantilla);           if (devolver != false) {
                System.out.println("Error al guardar el codigo de plantilla en la tabla usuario en
LoadPlantillaAction");           }           unaPersona.setCodigoPlantilla(codigoPlantilla);
            unaPersona.setUsuario(usuario);           session.setAttribute("unaPersona", unaPersona);        }
        return mapping.findForward("success");    }}
```

UsarPlantillaAction

```
package com.bvg.action;
import com.bvg.bean.Persona;
import com.bvg.bo.PersonaBO;
import com.bvg.formulario.PersonaForm;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionError;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
```

```
public class UsarPlantillaAction extends Action {
   public UsarPlantillaAction() {    }
   public ActionForward perform(ActionMapping mapping, ActionForm form,
                          HttpServletRequest request,
                          HttpServletResponse response) throws IOException,
                                                   ServletException {
      ActionErrors errors = new ActionErrors();        PersonaForm personaForm = (PersonaForm)form;
      String nombre = personaForm.getNombre();        if (nombre.trim().equals("")) {
         ActionError              error              =              new              ActionError("error.missing.nombre");
errors.add(ActionErrors.GLOBAL_ERROR, error);        }
      String apellido = personaForm.getApellido();        if (apellido.trim().equals("")) {
         ActionError error = new ActionError("error.missing.apellido");
         errors.add(ActionErrors.GLOBAL_ERROR, error);        }
      String usuario = personaForm.getUsuario();        if (usuario.trim().equals("")) {
         ActionError error = new ActionError("error.missing.usuario");
         errors.add(ActionErrors.GLOBAL_ERROR, error);        }
      String clave = personaForm.getClave();
      if (clave.trim().equals("")) {           ActionError error = new ActionError("error.missing.clave");
         errors.add(ActionErrors.GLOBAL_ERROR, error);        }
      if (errors.size() > 0) {           saveErrors(request, errors);
         return new ActionForward(mapping.getInput());        } else {
         PersonaBO unaPersonaBO = new PersonaBO();
         Persona unaPersona = new Persona();
         unaPersona.setNombre(personaForm.getNombre());
         unaPersona.setApellido(personaForm.getApellido());
         unaPersona.setUsuario(personaForm.getUsuario());
         unaPersona.setClave(personaForm.getClave());
         //unaPersonaBO.guardar(unaPersona);
         request.setAttribute("unaPersona", unaPersona);        }
      return mapping.findForward("success");    }
}
```

ValidaUsuarioAction
Trabaja con: EstiloPropio.jsp

Librerías :

```
package com.bvg.action;
import com.bvg.bean.Persona;
import com.bvg.bo.PersonaBO;
import com.bvg.formulario.PersonaForm;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionError;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
public class ValidaUsuario extends Action {
   public PersonaBO unaPersonaBO = new PersonaBO();
   public Persona unaPersona = new Persona();
   public ValidaUsuario() {    }
   public ActionForward perform(ActionMapping mapping, ActionForm form,
                          HttpServletRequest request,
```

```
                        HttpServletResponse response) throws IOException,
                         ServletException {        boolean esValido;        String esPlantilla;
        ActionErrors errors = new ActionErrors();
        PersonaForm personaForm = (PersonaForm)form;
        String usuario = personaForm.getUsuario();
        if (usuario.trim().equals("")) {
            ActionError error = new ActionError("error.missing.usuario");
            errors.add(ActionErrors.GLOBAL_ERROR, error);        }
        String clave = personaForm.getClave();        if (clave.trim().equals("")) {
            ActionError error = new ActionError("error.missing.clave");
            errors.add(ActionErrors.GLOBAL_ERROR, error);        }
        if (errors.size() > 0) {            //saveErrors(request, errors);
            return new ActionForward(mapping.getInput());        } else {
            unaPersona.setUsuario(personaForm.getUsuario());
            unaPersona.setClave(personaForm.getClave());
            esValido = unaPersonaBO.validaUsuario(usuario, clave);
            esPlantilla = unaPersonaBO.validaUsuario2(usuario, clave);
            int codigoPlantilla =
                unaPersonaBO.buscarCodigoPlantilla(usuario, clave);
            //personaForm.setPlantilla(esPlantilla);
            unaPersona.setArchivo(esPlantilla);
            unaPersona.setCodigoPlantilla(codigoPlantilla);        }
        if (esValido == true) {            System.out.println("Valor de variable esValido: " + esValido);
            System.out.println("Entro en VERDADERO"); HttpSession session = request.getSession(true);
            session.setAttribute("unaPersona", unaPersona);
            return  mapping.findForward("success");            }  else  {                  System.out.println("Entro  en
FALSO");            return mapping.findForward("failed");        }    }}
```

1.4. DESCRIPCION DE LA BASE DE DATOS

Tabla: Plantilla

Descripción: Esta tabla almacenará toda la información que el usuario constituya.

Fields	Type	Lenght	Primary	Forenks
Codigoplantilla	Int4		Si	
Nombre	Varchar	10		
Codigoetiqueta	Int4			Si

Codigosubtitulo	Int4			Si
Codigofondopagina	Int4			Si
Codigoimagenheader	Int4			Si
Codigotitulo	Int4			Si
Codigofondoformulario	Int4			Si
Codigolista	Int4			Si
Codigoborde	Int4			Si
Codigoboton	Int4			Si
Codigocombo	Int4			Si
Codigoimagenbody	Int4			Si
Codigoimagenfooter	Int4			Si

Tabla: ColorFondo

Descripción: Esta tabla almacenará toda la información del color de fondo.

Fields	Type	Lenght	Primary	Forenks
Codigocolorfondo	Varchar	10	Si	
Descripción	Varchar	10		
Rgb	Varchar	15		

Tabla: ColorFuente

Descripción: Esta tabla almacenará toda la información del color de la fuente.

Fields	Type	Lenght	Primary	Forenks
Codigocolorfuente	Varchar	10	Si	
Nombrecolor	Varchar	30		
Rgb	Varchar	10		

Tabla: FondoFormulario

Descripción: Esta tabla almacenará toda la información del fondo del formulario.

Fields	Type	Lenght	Primary	Forenks
Codigo fondoformulario	Int4		Si	
Codigocolor	Varchar	10		Si

Tabla: FondoPagina

Descripción: Esta tabla almacenará toda la información del fondo de la página.

Fields	Type	Lenght	Primary	Forenks
Codigofondopagina	Int4		Si	
Codigocolor	Varchar	10		Si
Codigoimagen	Int4	10		Si

Tabla: Imagen

Descripción: Esta tabla almacenará toda la información de la imagen.

Fields	Type	Lenght	Primary	Forenks
Codigoimagen	Varchar	10	Si	
Descripción	Varchar	10		
Archivo	Varchar	10		

Tabla: Plantillacodigo

Descripción: Esta tabla almacenará toda la información de los codigos de las tablas que conforman el estilo de la plantilla.

Fields	Type	Lenght	Primary	Forenks
Codigoplantilla	Varchar	10	Si	
Nombre	Varchar	10		
Codigoetiqueta	Varchar	10		Si
Codigotitulo	Varchar	10		Si
Codigosubtitulo	Varchar	10		Si
Codigofondopagina	Varchar	10		Si
Codigofondoformulario	Varchar	10		Si
codigoImagen	Varchar	10		Si

Tabla: Subtitulo

Descripción: Esta tabla almacenará toda la información del subtitulo.

Fields	Type	Lenght	Primary	Forenks
Codigosubtitulo	Int4		Si	
Codigotipofuente	Varchar	10		Si

Codigocolorfuente	Varchar	10		Si
Codigotamaniofuente	Varchar	10		Si
Estilofuente	Varchar	20		Si

Tabla: TamanioFuente

Descripción: Esta tabla almacenará toda la información del tamaño de la fuente.

Fields	Type	Lenght	Primary	Forenks
Codigotamaniofuente	Varchar	10	Si	
Tamaniofuente	Varchar	3		

Tabla: TipoFuente

Descripción: Esta tabla almacenará toda la información del tipo de la fuente.

Fields	Type	Lenght	Primary	Forenks
Codigotipofuente	Varchar	10	Si	
Tipofuente	Varchar	30		

Tabla: Titulo

Descripción: Esta tabla almacenará toda la información del titulo.

Fields	Type	Lenght	Primary	Forenks
Codigotitulo	Int4		Si	
Codigocolorfuente	Varchar	10		Si
Codigotipofuente	Varchar	10		Si
Codigotamaniofuente	Varchar	10		Si
Estilofuente	Varchar	10		Si

Tabla: Boton

Descripción: Esta tabla almacenará toda la información del botón.

Fields	Type	Lenght	Primary	Forenks
Codigoboton	Int4		Si	
Codigocolor	Varchar	10		Si
Iconoaceptar	Varchar	10		
Iconosalir	Varchar	10		

Iconover	Varchar	10		
Iconoregresar	Varchar	10		
iconocancelar	Varchar	10		
Iconoguardar	Varchar	10		
Iconobuscar	Varchar	10		
Codigotipofuente	Varchar	10		Si
Codigocolorfuente	Varchar	10		Si

Tabla: Combo

Descripción: Esta tabla almacenará toda la información del combo.

Fields	Type	Lenght	Primary	Forenks
Codigocombo	Int4		Si	
Codigocolorfondo	Varchar	10		Si
Codigotipofuente	Varchar	10		Si
Codigocolorfuente	Varchar	10		Si

Tabla: Etiqueta

Descripción: Esta tabla almacenará toda la información de la etiqueta.

Fields	Type	Lenght	Primary	Forenks
Codigoetiqueta	Int4		Si	
Codigotipofuente	Varchar	10		Si
Codigocolorfuente	Varchar	10		Si
Codigotamaniofuente	Varchar	10		Si
Estilofuente	Varchar	30		

Tabla: Boton

Descripción: Esta tabla almacenará toda la información del boton.

Fields	Type	Lenght	Primary	Forenks
Codigousuario	Varchar	5	Si	
Codigotipousuario	Varchar	5		Si
Usuario	Varchar	50		
Claveanterior	Varchar	15		
Claveactual	Varchar	15		

Pregunta	Varchar	100		
Respuesta	Varchar	100		
Estadousuario	Varchar	1		
Codigopersona	Int4			
Plantillacss	Varchar	150		
Codigoplantilla	Int4			Si

1.5. SCRIPT DE LA BASE DE DATOS

-- Table: usuarios

```
-- DROP TABLE usuarios;

CREATE TABLE usuarios
(
  codigousuario varchar(5) NOT NULL,
  codigotipousuario varchar(5),
  usuario varchar(50),
  claveanterior varchar(15),
  claveactual varchar(15),
  pregunta varchar(100),
  respuesta varchar(100),
  estadousuario varchar(1),
  codigopersona int4,
  plantillacss varchar(150),
  codigoplantilla int4,
  CONSTRAINT usuarios_pkey PRIMARY KEY (codigousuario),
  CONSTRAINT usuarios_codigotipousuario_fkey FOREIGN KEY (codigotipousuario)
      REFERENCES tipousuario (codigotipousuario) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT usuarios_codigousuario_fkey FOREIGN KEY (codigousuario)
```

```
    REFERENCES usuarios (codigousuario) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE usuarios OWNER TO postgres;
```

-- Table: titulo

```
-- DROP TABLE titulo;

CREATE TABLE titulo
(
  codigotitulo int4 NOT NULL,
  codigocolorfuente varchar(10),
  codigotipofuente varchar(10),
  codigotamaniofuente varchar(10),
  estilofuente varchar(10),
  CONSTRAINT codigotitulo PRIMARY KEY (codigotitulo),
  CONSTRAINT titulo_codigocolorfuente_fkey FOREIGN KEY (codigocolorfuente)
    REFERENCES colorfuente (codigocolorfuente) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT titulo_codigotamaniofuente_fkey FOREIGN KEY (codigotamaniofuente)
    REFERENCES tamaniofuente (codigotamaniofuente) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT titulo_codigotipofuente_fkey FOREIGN KEY (codigotipofuente)
    REFERENCES tipofuente (codigotipofuente) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE titulo OWNER TO postgres;
```

-- Table: tipofuente

```
-- DROP TABLE tipofuente;

CREATE TABLE tipofuente
(
  codigotipofuente varchar(10) NOT NULL,
  tipofuente varchar(30),
  CONSTRAINT codigotipofuente PRIMARY KEY (codigotipofuente)
)
WITHOUT OIDS;
ALTER TABLE tipofuente OWNER TO postgres;
```

-- Table: subtitulo

```
-- DROP TABLE subtitulo;

CREATE TABLE subtitulo
( codigosubtitulo int4 NOT NULL,
  codigotipofuente varchar(10),
  codigocolorfuente varchar(10),
  codigotamaniofuente varchar(10),
  estilofuente varchar(20),
  CONSTRAINT codigosubtitulo PRIMARY KEY (codigosubtitulo),
  CONSTRAINT subtitulo_codigocolorfuente_fkey FOREIGN KEY (codigocolorfuente)
      REFERENCES colorfuente (codigocolorfuente) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT subtitulo_codigotamaniofuente_fkey FOREIGN KEY (codigotamaniofuente)
      REFERENCES tamaniofuente (codigotamaniofuente) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT subtitulo_codigotipofuente_fkey FOREIGN KEY (codigotipofuente)
      REFERENCES tipofuente (codigotipofuente) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT
)WITHOUT OIDS;
ALTER TABLE subtitulo OWNER TO postgres;
```

-- Table: subtitulo

```
-- DROP TABLE subtitulo;

CREATE TABLE subtitulo
(
  codigosubtitulo int4 NOT NULL,
  codigotipofuente varchar(10),
  codigocolorfuente varchar(10),
  codigotamaniofuente varchar(10),
  estilofuente varchar(20),
  CONSTRAINT codigosubtitulo PRIMARY KEY (codigosubtitulo),
  CONSTRAINT subtitulo_codigocolorfuente_fkey FOREIGN KEY (codigocolorfuente)
      REFERENCES colorfuente (codigocolorfuente) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT subtitulo_codigotamaniofuente_fkey FOREIGN KEY (codigotamaniofuente)
      REFERENCES tamaniofuente (codigotamaniofuente) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT subtitulo_codigotipofuente_fkey FOREIGN KEY (codigotipofuente)
      REFERENCES tipofuente (codigotipofuente) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE subtitulo OWNER TO postgres;
```

-- Table: plantilla

```
-- DROP TABLE plantilla;

CREATE TABLE plantilla
(
  codigoplantilla int4 NOT NULL,
  codigoetiqueta int4,  codigosubtitulo int4,  codigofondopagina int4,
  codigoimagenheader int4,  codigotitulo int4,  codigofondoformulario int4,
  codigolista int4,  codigoborde int4,  codigoboton int4,
  codigocombo int4,
  nombre varchar(30),
  codigoimagenbody int4,
  codigoimagenfooter int4,
  codigoimagenfoto int4,
  CONSTRAINT plantilla_pkey PRIMARY KEY (codigoplantilla),
  CONSTRAINT plantilla_codigoboton_fkey FOREIGN KEY (codigoboton)
     REFERENCES boton (codigoboton) MATCH SIMPLE
     ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT plantilla_codigocombo_fkey FOREIGN KEY (codigocombo)
     REFERENCES combo (codigocombo) MATCH SIMPLE
     ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT plantilla_codigoetiqueta_fkey FOREIGN KEY (codigoetiqueta)
     REFERENCES etiqueta (codigoetiqueta) MATCH SIMPLE
     ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT plantilla_codigofondoformulario_fkey FOREIGN KEY (codigofondoformulario)
     REFERENCES fondoformulario (codigofondoformulario) MATCH SIMPLE
     ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT plantilla_codigofondopagina_fkey FOREIGN KEY (codigofondopagina)
     REFERENCES fondopagina (codigofondopagina) MATCH SIMPLE
     ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT plantilla_codigoimagen_fkey FOREIGN KEY (codigoimagenheader)
     REFERENCES imagen (codigoimagen) MATCH SIMPLE
     ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT plantilla_codigoimagenbody_fkey FOREIGN KEY (codigoimagenbody)
     REFERENCES imagen (codigoimagen) MATCH SIMPLE
     ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT plantilla_codigoimagenfooter_fkey FOREIGN KEY (codigoimagenfooter)
     REFERENCES imagen (codigoimagen) MATCH SIMPLE
     ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT plantilla_codigolista_fkey FOREIGN KEY (codigolista)
     REFERENCES listas (codigolista) MATCH SIMPLE
     ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT plantilla_codigosubtitulo_fkey FOREIGN KEY (codigosubtitulo)
     REFERENCES subtitulo (codigosubtitulo) MATCH SIMPLE
     ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT plantilla_codigotitulo_fkey FOREIGN KEY (codigotitulo)
     REFERENCES titulo (codigotitulo) MATCH SIMPLE
     ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
```

-- Table: listas

```
-- DROP TABLE listas;

CREATE TABLE listas
(
  codigolista int NOT NULL,
  codigocolor varchar(10),
  codigocolorfuente varchar(10),
  codigotamaniofuente varchar(10),
  codigoipofuente varchar(10),
  CONSTRAINT codigolistas PRIMARY KEY (codigolista),
  CONSTRAINT listas_codigocolorfuente_fkey FOREIGN KEY (codigocolorfuente)
      REFERENCES colorfuente (codigocolorfuente) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE listas OWNER TO postgres;
```

-- Table: imagen

```
-- DROP TABLE imagen;

CREATE TABLE imagen
(
  codigoimagen int4 NOT NULL,
  descripcion varchar(150),
  archivo bytea,
  CONSTRAINT codigoimagen PRIMARY KEY (codigoimagen)
)
WITHOUT OIDS;
ALTER TABLE imagen OWNER TO postgres;
```

-- Table: fondopagina

```
-- DROP TABLE fondopagina;
CREATE TABLE fondopagina
( codigofondopagina int4 NOT NULL,
  codigocolor varchar(10),
  codigoimagen int4,
  CONSTRAINT codigofondopagina PRIMARY KEY (codigofondopagina),
  CONSTRAINT fondopagina_codigocolor_fkey FOREIGN KEY (codigocolor)
      REFERENCES colorfondo (codigocolor) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT,
```

```
  CONSTRAINT fondopagina_codigoimagen_fkey FOREIGN KEY
(codigoimagen)
      REFERENCES imagen (codigoimagen) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE fondopagina OWNER TO postgres;
```

-- Table: fondoformulario

```
-- DROP TABLE fondoformulario;

CREATE TABLE fondoformulario
(
  codigofondoformulario int4 NOT NULL,
  codigocolor varchar(10),
  CONSTRAINT codigofondoformulario PRIMARY KEY (codigofondoformulario),
  CONSTRAINT fondoformulario_codigocolor_fkey FOREIGN KEY (codigocolor)
      REFERENCES colorfondo (codigocolor) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE fondoformulario OWNER TO postgres;
```

-- Table: etiqueta

```
-- DROP TABLE etiqueta;

CREATE TABLE etiqueta
(
  codigoetiqueta int NOT NULL,
  codigotipofuente varchar(10),
  codigocolorfuente varchar(10),
  codigotamaniofuente varchar(10),
  estilofuente varchar(30),
  CONSTRAINT codigoetiqueta PRIMARY KEY (codigoetiqueta),
  CONSTRAINT codigocolorfuente FOREIGN KEY (codigocolorfuente)
      REFERENCES colorfuente (codigocolorfuente) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT codigotamaniofuente FOREIGN KEY (codigotamaniofuente)
      REFERENCES tamaniofuente (codigotamaniofuente) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT codigotipofuente FOREIGN KEY (codigotipofuente)
      REFERENCES tipofuente (codigotipofuente) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT etiqueta_codigocolorfuente_fkey FOREIGN KEY (codigocolorfuente)
      REFERENCES colorfuente (codigocolorfuente) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT etiqueta_codigotamaniofuente_fkey FOREIGN KEY (codigotamaniofuente)
      REFERENCES tamaniofuente (codigotamaniofuente) MATCH SIMPLE
```

```
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT etiqueta_codigotipofuente_fkey FOREIGN KEY (codigotipofuente)
    REFERENCES tipofuente (codigotipofuente) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE etiqueta OWNER TO postgres;
```

-- Table: combo

```
-- DROP TABLE combo;

CREATE TABLE combo
(
  codigocombo int4 NOT NULL,
  codigocolorfondo varchar,
  codigotipofuente varchar,
  codigocolorfuente varchar,
  CONSTRAINT codigocombo PRIMARY KEY (codigocombo),
  CONSTRAINT combo_codigocolorfondo_fkey FOREIGN KEY (codigocolorfondo)
    REFERENCES colorfondo (codigocolor) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT combo_codigocolorfuente_fkey FOREIGN KEY (codigocolorfuente)
    REFERENCES colorfuente (codigocolorfuente) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT combo_codigotipofuente_fkey FOREIGN KEY (codigotipofuente)
    REFERENCES tipofuente (codigotipofuente) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE combo OWNER TO postgres;
```

-- Table: colorfondo

```
-- DROP TABLE colorfondo;

CREATE TABLE colorfondo
(
  codigocolor varchar(10) NOT NULL,
  descripcion varchar(10),
  rgb varchar(15),
  CONSTRAINT "codigoColor" PRIMARY KEY (codigocolor)
)
WITHOUT OIDS;
ALTER TABLE colorfondo OWNER TO postgres;
```

-- Table: colorfuente

```
-- DROP TABLE colorfuente;


CREATE TABLE colorfuente
(
  codigocolorfuente varchar(10) NOT NULL,
  nombrecolor varchar(30),
  rgb varchar(10),
  CONSTRAINT codigocolorfuente PRIMARY KEY (codigocolorfuente)
)
WITHOUT OIDS;
ALTER TABLE colorfuente OWNER TO postgres;
```

CREATE TABLE boton

```
(
  codigoboton int4 NOT NULL,
  codigocolor varchar(10),
  codigotipofuente varchar(10),
  codigocolorfuente varchar(10),
  CONSTRAINT codigoboton PRIMARY KEY (codigoboton),
  CONSTRAINT boton_codigocolor_fkey FOREIGN KEY (codigocolor)
      REFERENCES colorfondo (codigocolor) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT boton_codigocolorfuente_fkey FOREIGN KEY (codigocolorfuente)
      REFERENCES colorfuente (codigocolorfuente) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT boton_codigotipofuente_fkey FOREIGN KEY (codigotipofuente)
      REFERENCES tipofuente (codigotipofuente) MATCH SIMPLE
      ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE boton OWNER TO postgres;
```

1.6.DICCIONARIO DE DATOS

Variable	Type	Description
codigoNuevaEtiqueta	Int	Esta variable almacena el código de la nueva etiqueta.
codigoNuevoTitulo	Int	Esta variable almacena el código del nuevo título.
codigoNuevoBoton	Int	Esta variable almacena el código de la nuevo botón.
codigoNuevoFondoFormulario	Int	Esta variable almacena el código de la nuevo fondo de formulario.
codigoNuevoFondoPagina	Int	Esta variable almacena el código del nuevo fondo de página.
codigoNuevaLista	Int	Esta variable almacena el código de la nueva lista.
codigoNuevoCombo	Int	Esta variable almacena el código del nuevo combo.
banderaNuevaEtiqueta	String	Esta variable almacena la bandera de la nueva etiqueta.
banderaNuevoTitulo	String	Esta variable almacena la bandera del nuevo título.
banderaNuevoSubtitulo	String	Esta variable almacena la bandera del nuevo subtítulo.

DICCIONARIO DE DATOS

Variable	Type	Description
banderaNuevoBoton	String	Esta variable almacena la bandera del nuevo botón.
banderaNuevoFondoPagina	String	Esta variable almacena la bandera del nuevo fondo de página.
banderaNuevoFondoFormulario	String	Esta variable almacena la bandera del nuevo fondo de formulario.
banderaNuevoCombo	String	Esta variable almacena la bandera del nuevo combo.
banderaNuevaLista	String	Esta variable almacena la bandera de la nueva lista.
valorEtiqueta	boolean	Esta variable almacena .T. o .F. del valor etiqueta.
valorTitulo	boolean	Esta variable almacena .T. o .F. del valor del título.
valorSubtitulo	boolean	Esta variable almacena .T. o .F. del valor del subtítulo.
valorBoton	boolean	Esta variable almacena .T. o .F. del valor del botón.
valorFondoPagina	boolean	Esta variable almacena .T. o .F. del valor del fondo de página.

DICCIONARIO DE DATOS

Variable	Type	Description
valorFondoFormulario	Boolean	Esta variable almacena .T. o .F. del valor del fondo del formulario.
valorCombo	Boolean	Esta variable almacena .T. o .F. del valor del combo.
valorLista	Boolean	Esta variable almacena la bandera del nuevo valor de lista.
Devolver	Boolean	Esta variable devuelve .T. o .F. si el usuario ha guardado la plantilla.
Esvalido	Boolean	Esta variable devuelve .T. o .F. si el usuario existe o no en la base de datos.
Usuario	String	Esta variable almacena el nombre del usuario.
Banderacrear	String	Esta variable almacena el valor de 1 o 0, según la plantilla que se vaya a crear.
codigoColorFondo	String	Esta variable almacena el código del color de fondo.
codigoTipoFuente	String	Esta variable almacena el código del color de tipo de fuente.
codigoColorFuente	String	Esta variable almacena el código del color de la fuente.

DICCIONARIO DE DATOS

Variable	Type	Description
nombrecolor	String	Esta variable almacena el nombre del color de la fuente.
rgb	String	Esta variable almacena el valor hexadecimal de los colores.
codigotamaniofuente	String	Esta variable almacena el código del tamaño de la fuente
tamaniofuente	String	Esta variable almacena el valor del tamaño de la fuente.
codigotipofuente	String	Esta variable almacena el código del tipo de la fuente
tipofuente	String	Esta variable almacena el valor del tipo de la fuente.
codigocolorfondo	String	Esta variable almacena el valor del código del color de fondo.
descripcion	String	Esta variable almacena el nombre del color de fondo.
clave	String	Esta variable almacena la clave del usuario.
nombre	String	Esta variable almacena el nombre del usuario.
apellido	String	Esta variable almacena el apellido del usuario.
archivo	byte	Esta variable almacena el nombre de la imagen a “subirse”.

DICCIONARIO DE DATOS

Variable	Type	Description
Codigofondopagina	Int	Esta variable almacena el código del color de fondo de la página.
codigoImagen	int	Esta variable almacena el código de la imagen.
Estilo	String	Esta variable almacena el estilo.
codigofondoformulario	Int	Esta variable almacena el código del fondo de formulario.
Imagen	String	Esta variable almacena el nombre de la imagen.
codigoImagenHeader	Int	Esta variable almacena el código de la cabecera.
codigoImagenBody	Int	Esta variable almacena el código del cuerpo.
codigoImagenFooter	Int	Esta variable almacena el código del pie de página.
codigoImagenFoto	Int	Esta variable almacena el código de la imagen de la foto.
buscarCodigo	Int	Esta variable busca un código en particular.
valorDevolver	vector	Variable de tipo vector, que devuelve un arreglo de datos.
nombrePlantilla	String	Variable que almacena el nombre de la plantilla a guardarse.

DICCIONARIO DE DATOS

Variable	Type	Description
ImagenPlantilla	String	Esta variable almacena el nombre de la imagen de la plantilla.
codigoPlantilla	int	Esta variable almacena el código de la plantilla.
tipofuenteEtiqueta	String	Variable que almacena el valor de tipo de fuente de la etiqueta.
colorfuenteEtiqueta	String	Variable que almacena el valor del color de la fuente de la etiqueta.
tamaniofuenteEtiqueta	String	Variable que almacena el valor del tamaño de la fuente de la etiqueta.
tipofuenteTitulo	String	Variable que almacena el valor de tipo de fuente del titulo.
colorfuenteTitulo	String	Variable que almacena el valor del color de la fuente del título.
tamaniofuenteTitulo	String	Variable que almacena el valor del tamaño de la fuente del título.
tipofuenteSubtitulo	String	Variable que almacena el valor de tipo de fuente del Subtitulo.
colorfuenteSubtitulo	String	Variable que almacena el valor del color de la fuente del Subtítulo.
tamaniofuenteSubtitulo	String	Variable que almacena el valor del tamaño de la fuente del Subtítulo.

DICCIONARIO DE DATOS

Variable	Type	Description
colorfondoPagina	String	Esta variable almacena el nombre del color de la fuente.
colorBorde	String	Esta variable almacena el valor de color de borde.
colorfondoFormulario	String	Esta variable almacena el valor de color de fondo del formulario.
imagenCabecera	FormFile	Esta variable almacena la imagen que va en la cabecera.
tipofuenteCombo	String	Variable que almacena el valor de tipo de fuente del Combo.
colorfuenteCombo	String	Variable que almacena el valor del color de la fuente del Combo.
colorfondoCombo	String	Variable que almacena el valor del color del fondo del Combo.
tipofuenteLista	String	Variable que almacena el valor de tipo de fuente de la Lista.
colorfuenteLista	String	Variable que almacena el valor del color de la fuente de la Lista.
colorfondoLista	String	Variable que almacena el valor del color del fondo de la Lista.

DICCIONARIO DE DATOS

Variable	Type	Description
tipofuenteBoton	String	Esta variable almacena el valor del tipo de fuente del botón.
colorfuenteBoton	String	Esta variable almacena el valor del color de fuente del botón.
fondofuenteBoton	String	Esta variable almacena el valor de fondo de la fuente del botón.
Bos	OutStream	Variable que sirve para escribir el archivo que se baja de la base de datos.
archivoFotoBase	Byte	Recibe el archivo en bytes que esta almacenado en la base de datos. Max:8192 bytes
Rutafoto	String	Almacena la ruta de la foto.

1.7. LIBRERIAS

import java.io.IOException;

import java.util.Vector;

import javax.servlet.ServletException;

import javax.servlet.ServletContext;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;

import org.apache.struts.action.Action;

import org.apache.struts.action.ActionError;

import org.apache.struts.action.ActionErrors;

import org.apache.struts.action.ActionForm;

import org.apache.struts.action.ActionForward;

import org.apache.struts.action.ActionMapping;

1.8. ESTADISTICAS DE ARCHIVOS USADOS EN EL MODULO DE APARIENCIA DINAMICA WEB.

Número de Archivos ext .JAVA: 61

Número de Archivos ext .JSP: 27

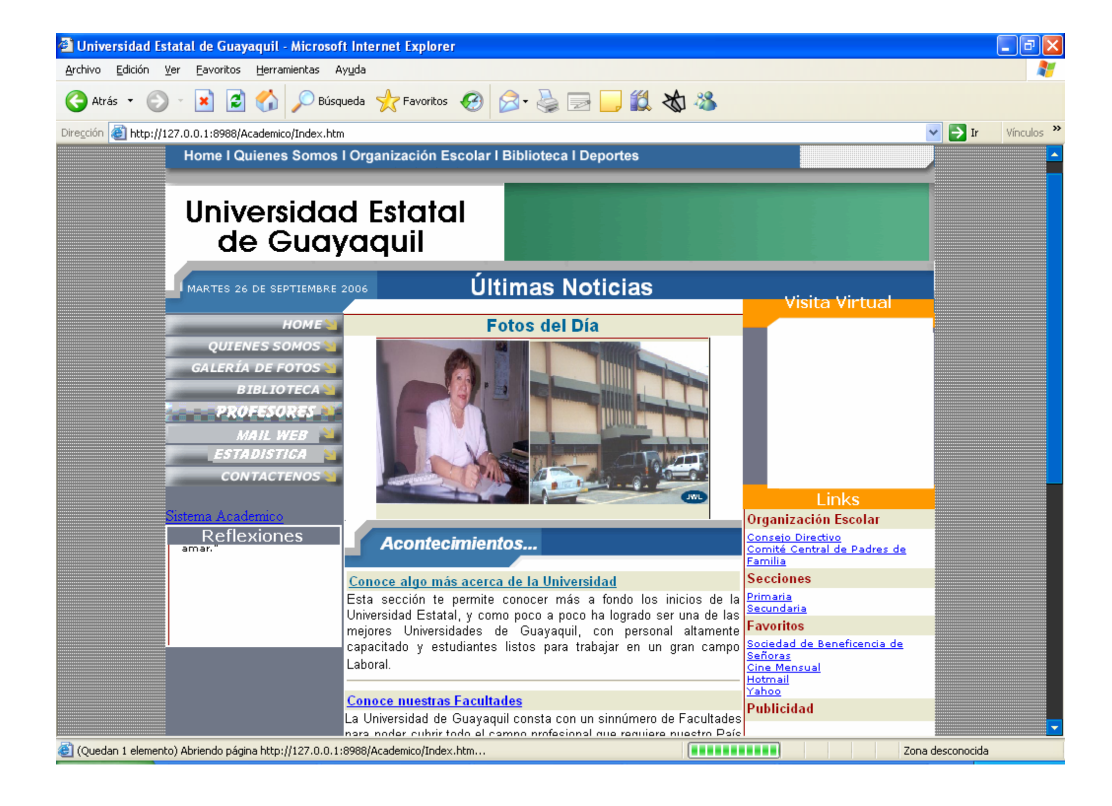
Número de Librerías: 15

CAPÍTULO 2

2. MANUAL DE USUARIO

2.1. FLUJO DE VENTANAS

Sitio de Acceso Web al Sistema



Descripción:

Esta pantalla describe el acceso al sitio web de nuestro SISTEMA ACADEMICO, el cual la describiremos por áreas:

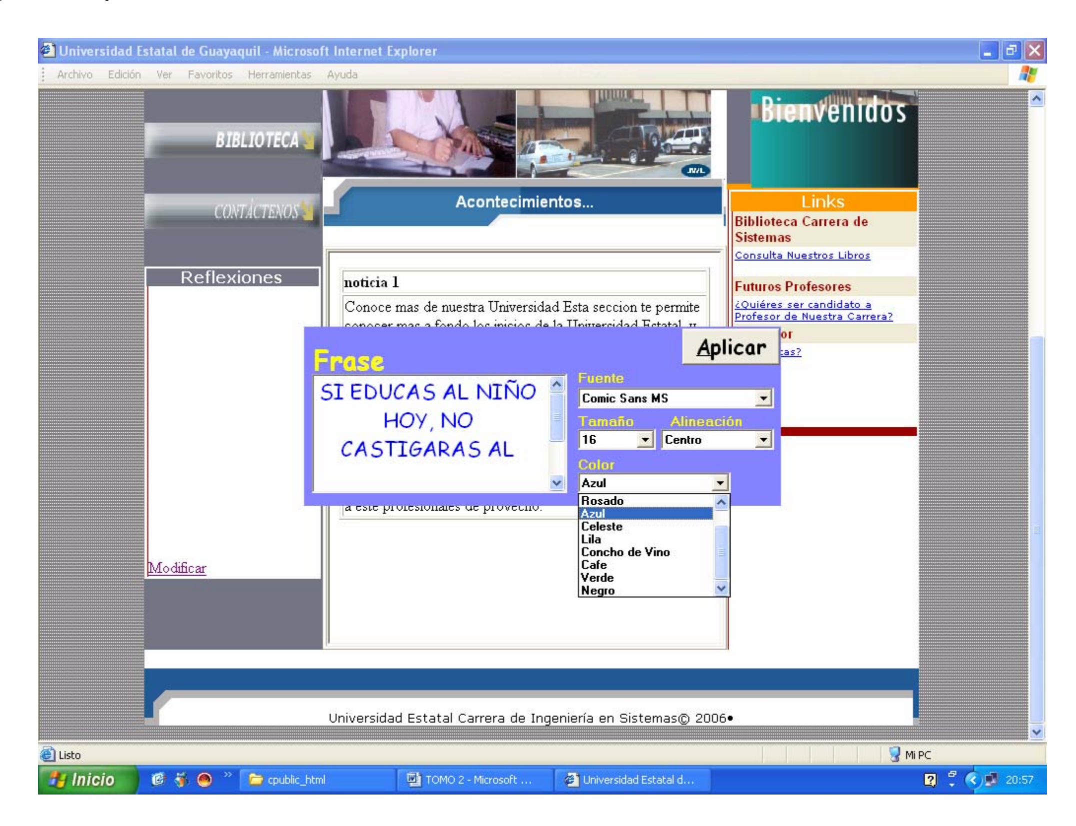
En su parte superior se describe la UNIVERSIDAD DE GUAYAQUIL y un menú de opciones como QUIENES SOMOS – HOME - CONTACTENOS .

En su parte izquierda tenemos un spot publicitario y debajo de él una leyenda interactiva de Reflexiones que puede ser personalizado, de la siguiente manera:



Para ello hay que ingresar con clave de **ADMINISTRADOR**, ya que es el único perfil que puede cambiar la interacción del sitio web, ya sea en su nota de reflexiones o fotos actualizadas.

La clave es **admin**, como referencia a Administrador, e ingresaría a la siguiente pantalla:



Esta describe una pantalla en la que se despliega un subproceso, donde puede escribir **LA NUEVA FRASE**, y detallar cual será su tipo de fuente, su tamaño, su color y la alineación que tendrá.

A continuación, pulsa el botón **APLICAR** y los cambios quedan hechos.



En su parte derecha, tenemos un despliegue de fotos de la Carrera de Ingeniería de Sistemas y debajo de él, Links de temas relacionados con la carrera.

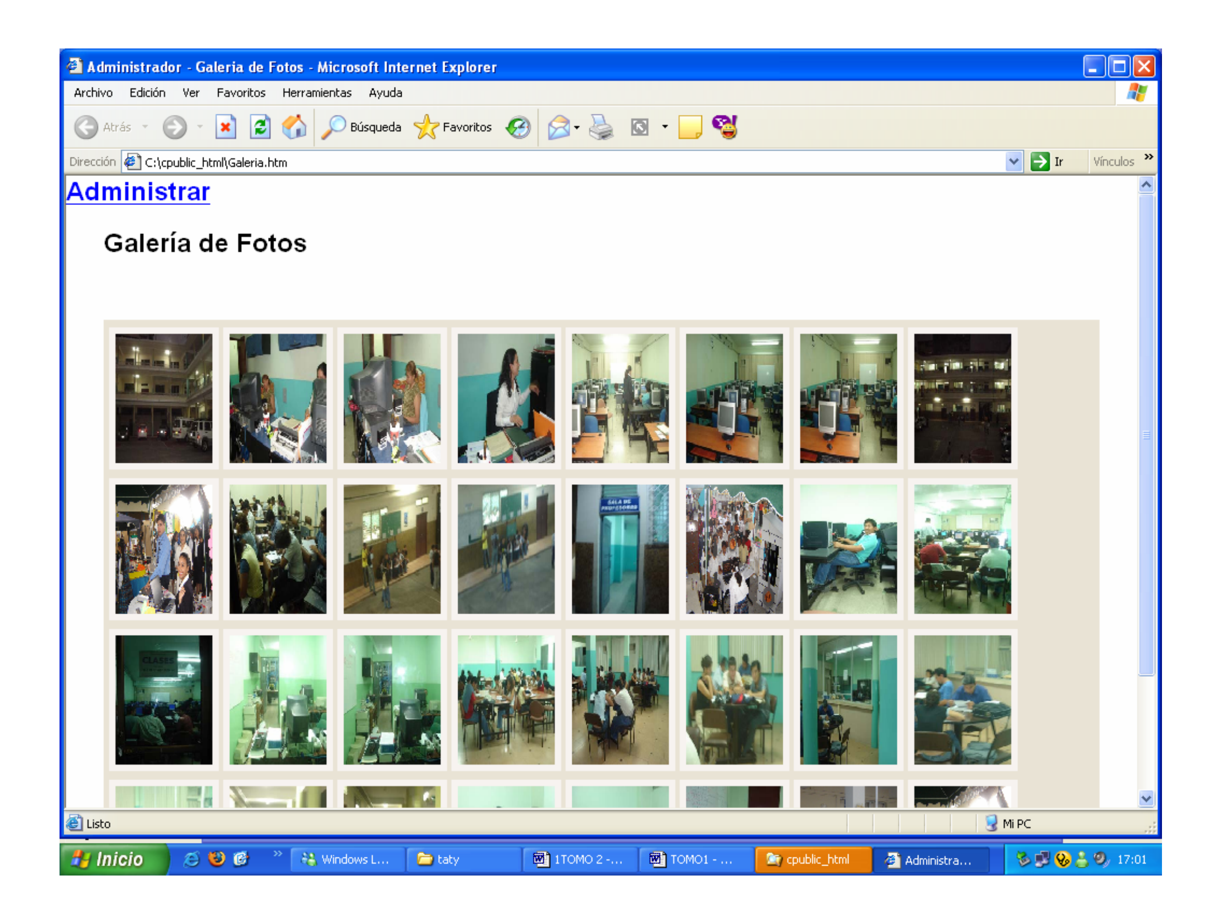
En su parte central, tenemos como **FOTO DEL DIA** a la Ing. Matilde Alvear, Directora de la Carrera y el Rectorado de la Universidad.

Debajo de él, noticias interactivas de lo que acontece en la carrera, ya sean eventos, conferencias, elecciones, y farándula.



Como pie de página, la descripción de propiedad intelectual del Sistema Académico.

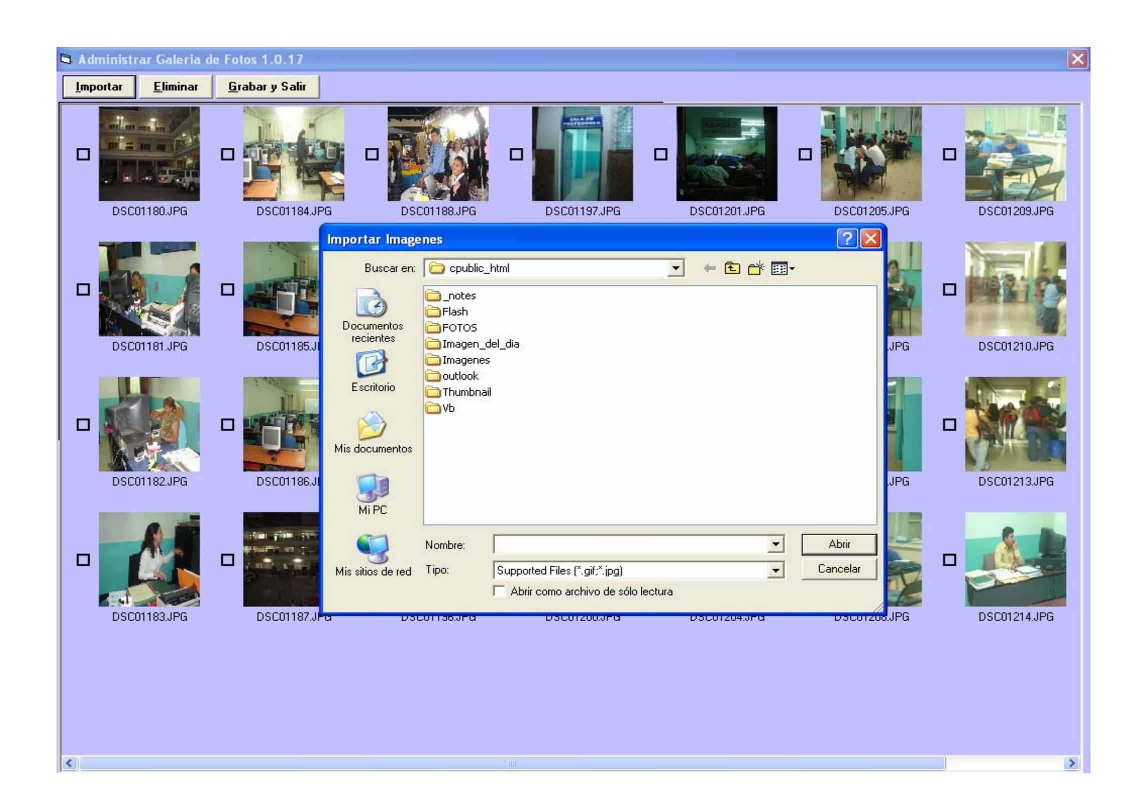
En la opción de **GALERIA DE FOTOS**, tenemos un sinnúmero de fotos que pueden ser reemplazadas por otras nuevas, para ir renovando la galería a gusto de quien la disfruta.



Al dar clic en el botón administrar me permitirá modificar, eliminar, reemplazar fotos



Si presiona el botón IMPORTAR, aparecerá una pantalla como la siguiente en donde podrá seleccionar desde cualquier dispositivo o carpeta de su computador la foto que desea el administrador publicar e inmediatamente aparecerá en la galería, a continuación para aceptar los cambios se da clic en el botón GUARDAR Y SALIR.

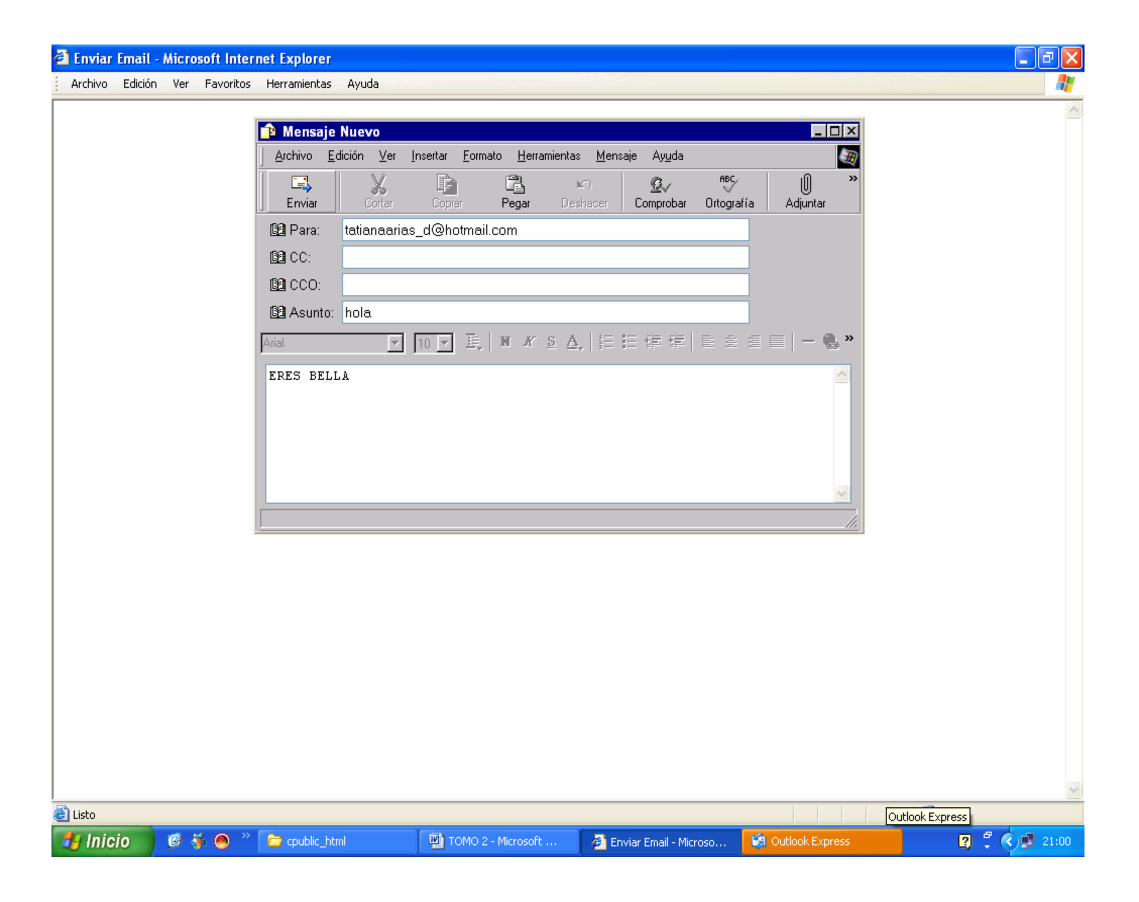


Para poder eliminar alguna foto se selecciona la o las fotos dando clic en el cuadrito que se encuentra a lado izquierdo de cada imagen.

Y así como en la opción de IMPORTAR, se presiona el botón GUARDAR Y SALIR. Y listo la galería de Imágenes quedará actualizada según las actividades que se realicen y que el administrador desee publicar.

En la opción de **CONTACTENOS**, tenemos un correo al cual usted puede enviarnos vuestras sugerencias, para estar siempre en contacto a sus inquietudes.

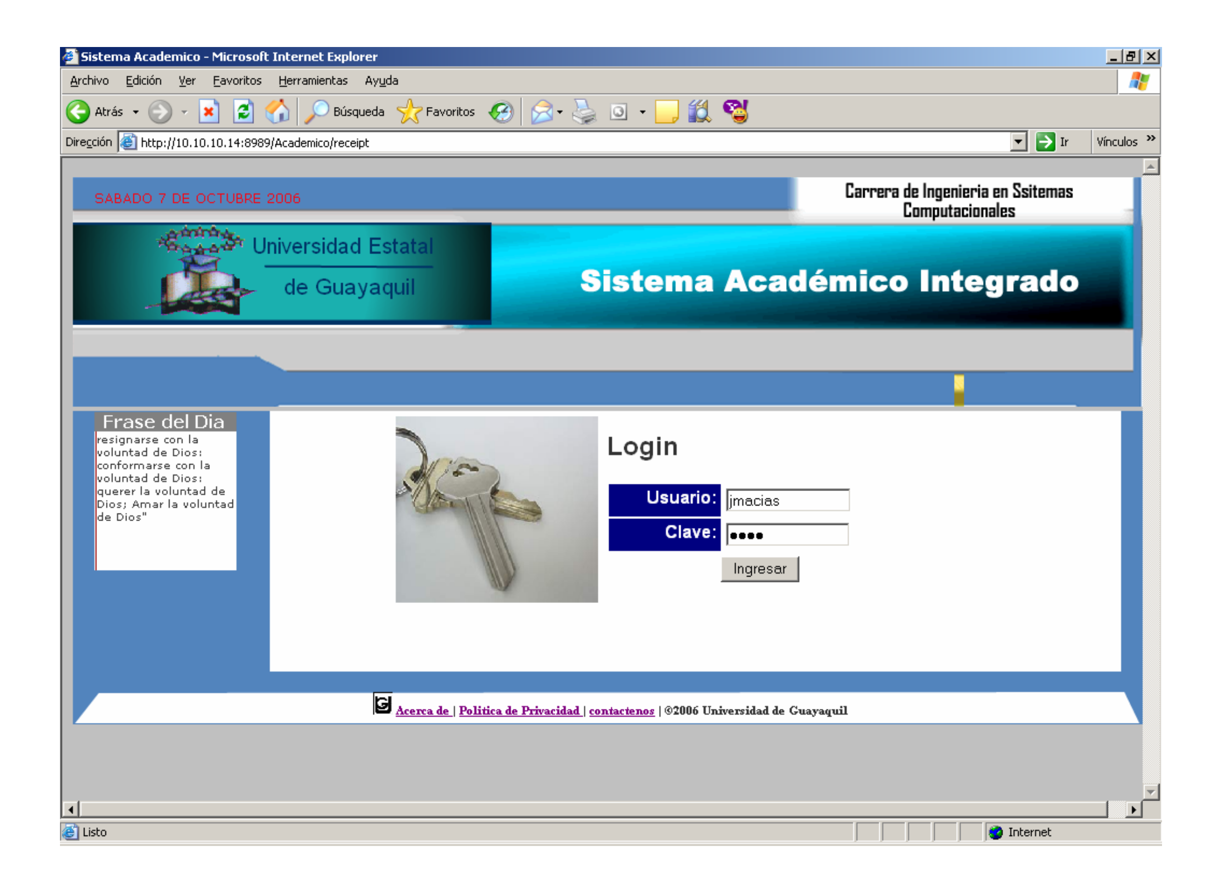
El email es: tatianaarias_d@hotmail.com



Esta simulación de la Página de la Universidad, también permite ingresar a otras páginas informativas importantes de esta loable institución

Pantalla de Acceso al Sistema

Al dar clic en el vínculo de la página general de la Universidad se vinculará con la página siguiente:



Descripción:

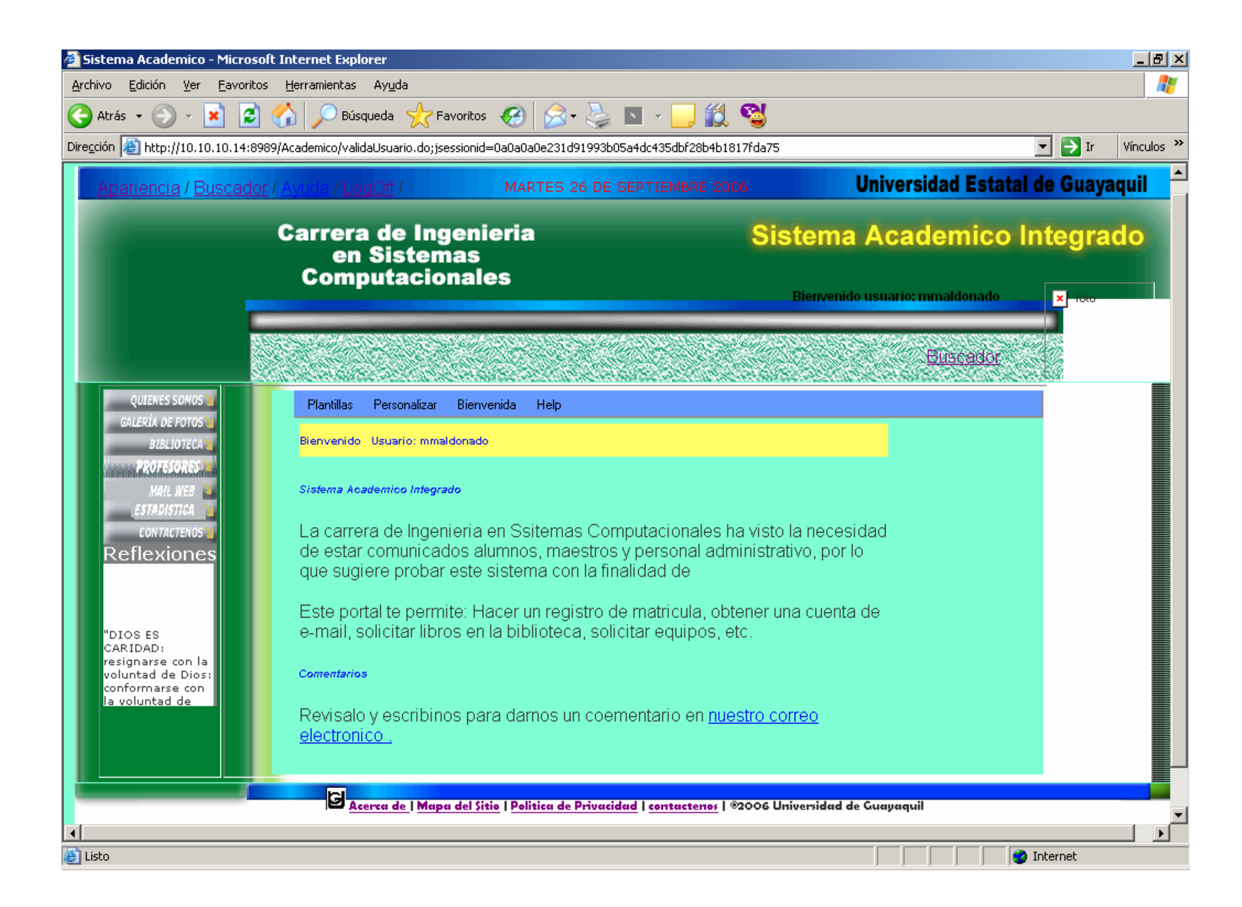
Esta pantalla consta de una cabecera donde se indica el titulo del sistema académico y la carrera de ingeniería de sistemas a la cual pertenece.

En su parte central se muestra la petición de Usuario y Clave el cual debe constar en la base de datos de Usuarios para poder acceder, mediante el botón ***Ingresar***.

En su parte izquierda, se muestra una serie de opciones (menú), a los distintos módulos como referencia a dichos tópicos.

En su parte inferior a manera de pie de página consta una leyenda que hace referencia a los derechos de Autor, en este caso la Universidad de Guayaquil sobre el Sistema.

Pantalla de Bienvenida al Sistema



Descripción:

Una vez que hemos ingresado con nuestro usuario y clave, se nos muestra una pantalla de bienvenida, el cual se distingue la fecha y el usuario que ingreso. Dicha pantalla consta de lo siguiente:

En su parte superior aparece una serie de opciones, como son:

- **Apariencia**: Se ingresa a las opciones de cambio de apariencia, como son el Plantilla y Estilo Propio.

- **Buscador**: Esta opción me permite realizar la búsqueda de un tema en particular.

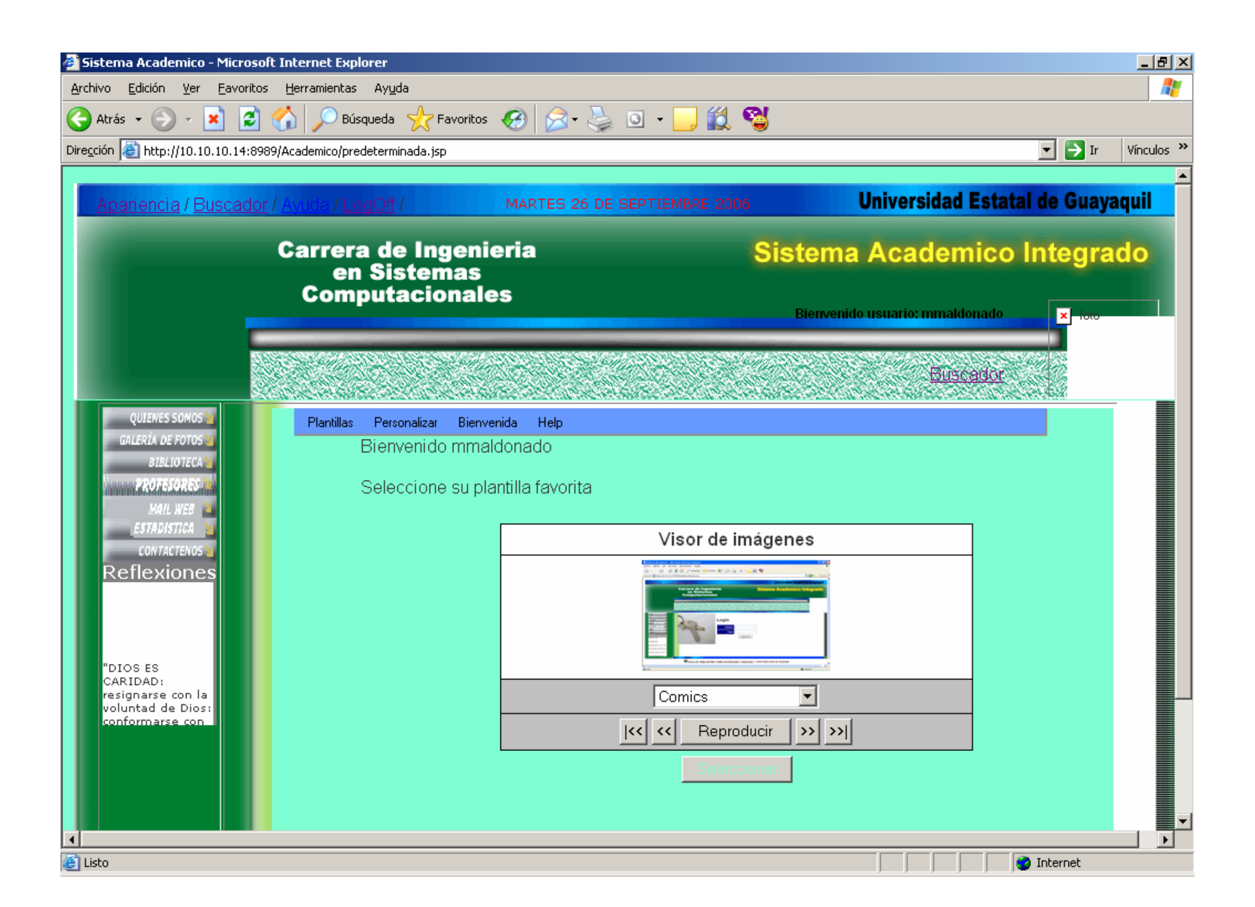
- **Ayuda**: Muestra la ayuda referente a los módulos que tiene el Sistema Académico.

- **LogOff**: Dicha opción realiza el cierre de sesión de un usuario.

De lado superior derecho, se reserva un espacio para la Foto del usuario o imagen que desee mostrar.

En su parte central, se muestra una barra de opciones como son las plantillas, para cambiar la apariencia; personalizar, para dar un toque personal y una breve explicación del porque de la creación del sistema en sí.

Pantalla de Opción de Plantillas

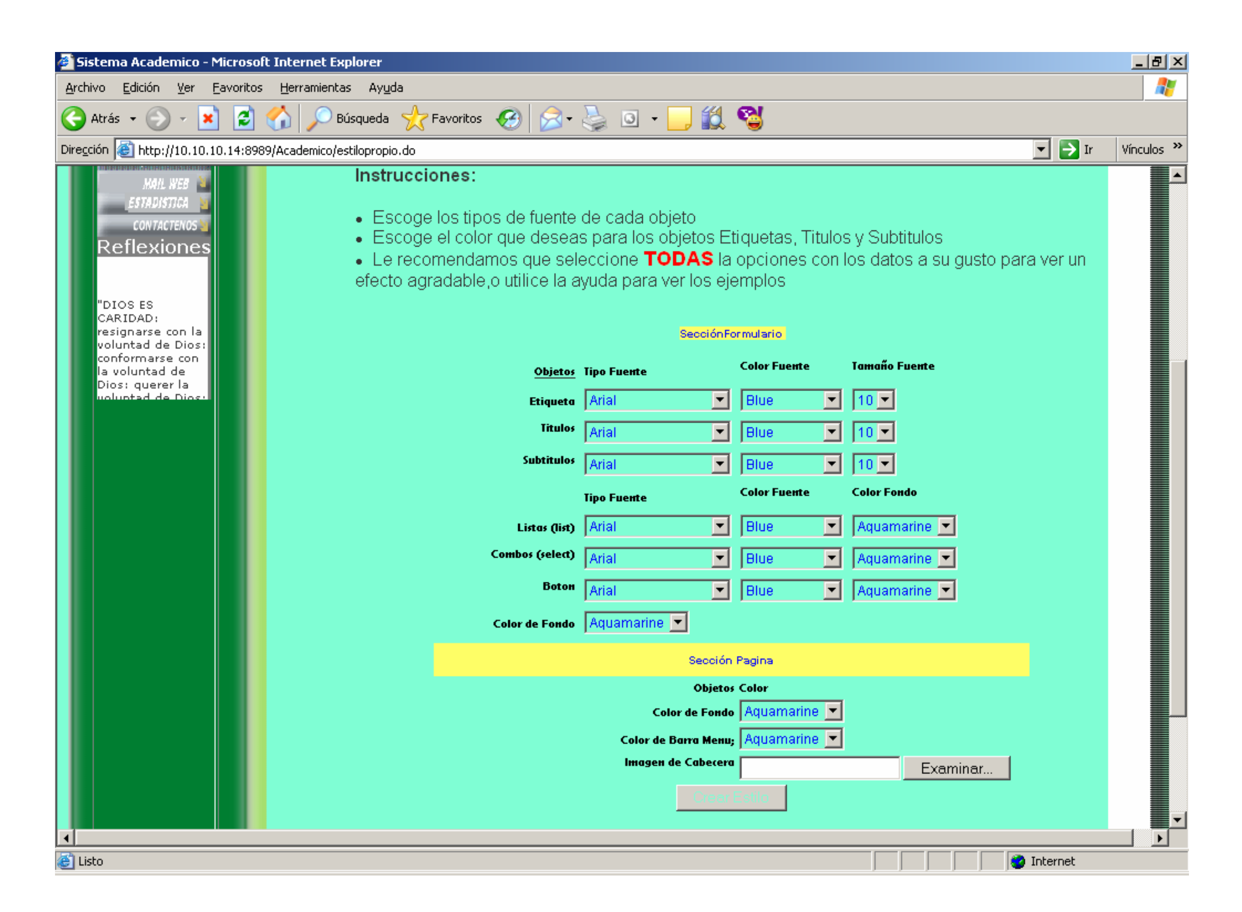


Descripción:

Esta pantalla describe la presentación de plantillas a escoger por el usuario, mediante un combo de selección. El usuario escoge la plantilla de su preferencia y pulsa el Botón Seleccionar y la pantalla cambiara su apariencia.

Tómese en cuenta que el menú de opciones y el pie de página siempre están a disposición del usuario.

Pantalla de Opción de Personalizado



Descripción:

Esta pantalla describe la personalización que puedo realizarle a mi plantilla, distinguiéndolo en 2 áreas:

1. **Área.** ***Sección Formulario***: Al hablar de la sección formulario, hablamos de la información central que el usuario en su parte central puede personalizar, tales como:

   Tipo de Fuente – Color de la fuente – Tamaño de la fuente, a elementos como:

   Etiqueta – Títulos – Subtítulos – Listas – Combos – Botones – Color de Fondo

2. **Área.** ***Sección Página***: Al hablar de la sección pagina, hablamos del contorno de la página a mostrarse.

   Es aquí donde el usuario puede personalizar:

   Color de fondo – Color de borde Menú – Ubicar una imagen de cabecera

La combinación de colores queda a criterio del usuario, pero se le recomienda usar colores suaves para los contornos y colores fuertes para resaltar algún titulo en particular.

Algo muy importante, es que los cambios que se realicen en esta sección de ser en TODAS las opciones, para que surta el efecto requerido.

CONCLUSIÓN

Creemos que la principal condición que debe cumplir un sitio web es el diseño armonioso y original. La mayoría de los diseñadores de páginas web se basan demasiado sobre el aspecto técnico, descuidando por completo el aspecto visual. Nosotros nos preocupamos por el aspecto visual tanto como por el técnico, asegurándonos de que su nuestro Sistema Académico Web será del agrado de sus visitantes y el nuestro propio. Se tiene páginas optimizadas para ser descargadas en tiempos realmente cortos sin sacrificar la calidad de las imágenes. Esto es importante, ya que la mayoría de las personas que visitan una página lenta, no terminan de ver por completo la información y concluyen repudiando el Sitio Web, y por ende, a sus productos o servicios.

Estamos preparados para los buscadores. ¿Cuántas personas tienen un Sitio Web y nadie que lo vaya a ver?. Un Sitio Web es inútil si nadie puede saber que existe. Una mala planificación de las páginas y la información que las componen se traducirá en buscadores que no reconocen su contenido o no listan las páginas con la importancia que su nuestra universidad y específicamente la Facultad requiere, además queremos mostrar un manual de usuario y técnico que permita realizar nuevas modificaciones o mejoras al sistema para beneficio de todo un colectivo, y que de esta manera tenga una mejor comprensión.